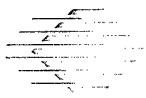


Carnegie Mellon
Software Engineering Institute

State of the Practice of Intrusion Detection Technologies

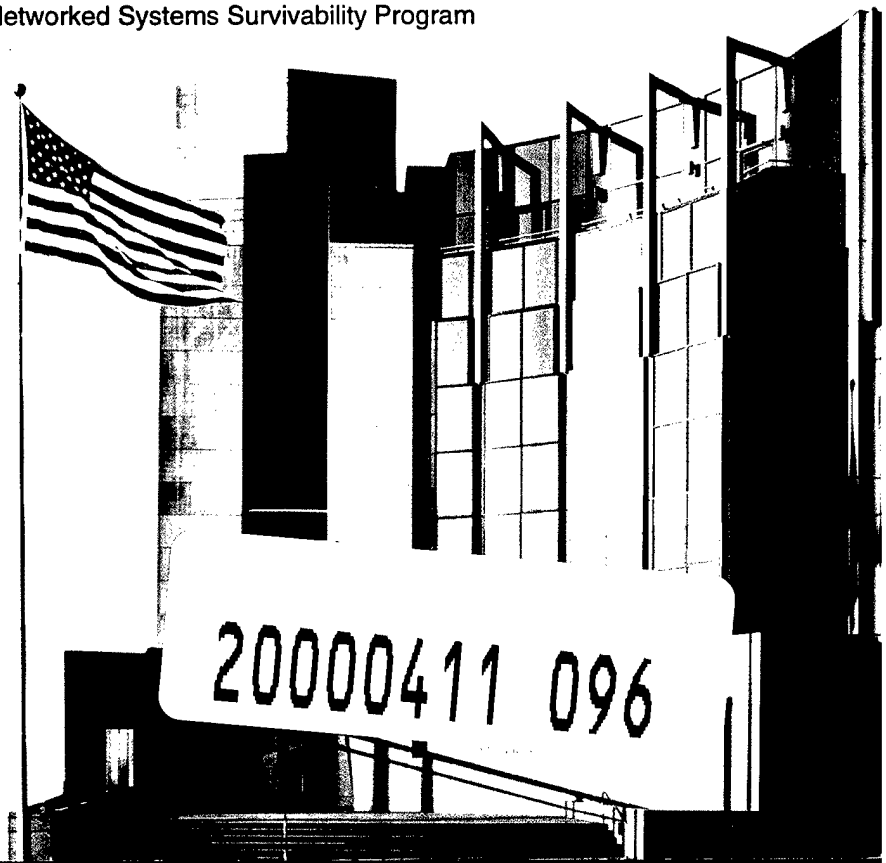
Julia Allen
Alan Christie
William Fithen
John McHugh
Jed Pickel
Ed Stoner

Contributors:
James Ellis
Eric Hayes
Jerome Marella
Bradford Willke

January 2000

Networked Systems Survivability Program

TECHNICAL REPORT
CMU/SEI-99-TR-028
ESC-99-028



Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of "Don't ask, don't tell, don't pursue" excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

Obtain general information about Carnegie Mellon University by calling (412) 268-2000.

Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

State of the Practice of Intrusion Detection Technologies

CMU/SEI-99-TR-028

ESC-99-028

Authors:
Julia Allen
Alan Christie
William Fithen
John McHugh
Jed Pickel
Ed Stoner

Contributors:
James Ellis
Eric Hayes
Jerome Marella
Bradford Willke

January 2000

Networked Systems Survivability Program

This work is sponsored by the Air Force Research Laboratory and the Air Force Computer Resources Support Improvement Program. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Please see http://www.sei.cmu.edu/publications/pubweb.html for information about how to order paper copies of SEI reports.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

Acknowledgments

This report was sponsored by the Air Force Research Laboratory (Mr. Dwayne Allain) and by the Air Force Computer Resources Support Improvement Program (Lt Col Joseph Jarzombek).

The authors acknowledge contributions made to this report, which were in the form of document production, interviews, and reviews of draft sections. The quality of the document was greatly enhanced by the expertise of the following individuals:

Ed Amoroso, AT&T
Jay Larrew, AFIWC (Air Force Information Warfare Center)
Teresa Lunt, Xerox
Roy Maxion, Carnegie Mellon University
Mark Woods, Internet Security Systems

Lincoln Laboratory:
Richard Lippman
Marc Zissman

MITRE:
Joshua Guttman
Leonard LaPadula
Marion Michaud
Jeffrey Picciotto

Software Engineering Institute:
Bunny Bernfeld
Claire Dixon
Mary Jonson
Tom Longstaff
Klaus-Peter Kossakowski
Jim Main
Mindi McDowell
Jacqueline Prause
Sheila Rosenthal
Tim Shimeall
Pam Williams

List of Figures

List of Tables

Executive Summary

Attacks on the nation's computer infrastructures are a serious problem. Over the past 12 years, the growing number of computer security incidents on the Internet has reflected the growth of the Internet itself. Because most deployed computer systems are vulnerable to attack, intrusion detection (ID) is a rapidly developing field. Intrusion detection is an important technology business sector as well as an active area of research.

Vendors make many claims for their products in the commercial marketplace so separating hype from reality can be a major challenge. A goal of this report is to provide an unbiased assessment of publicly available ID technology. We hope this will help those who purchase and use ID technology to gain a realistic understanding of its capabilities and limitations. The report raises issues that we believe are important for ID system (IDS) developers to address as they formulate product strategies. The report also points out relevant issues for the research community as they formulate research directions and allocate funds.

Implementing intrusion detection systems on networks and hosts requires a broad understanding of computer security. The complexity of information technology infrastructures is increasing beyond any one person's ability to understand them, let alone administer them in a way that is operationally secure. Vendors are rapidly releasing new ID systems and aggressively competing for market share in an expanding market. Many products started out as point solutions. However, in response to consumers' inability to fully understand and use many ID systems, vendors are attempting to integrate approaches to solve a broader range of computer security problems. Evaluating ID systems is non-trivial and there is a lack of credible, comprehensive product evaluation information. Hiring and retaining personnel to competently administer security in general and intrusion detection in particular are increasing challenges. All of this rapid change makes it very difficult for an organization to implement an effective, long-term security strategy.

After reviewing the surveys cited in this report, one could conclude that ID technologies are becoming an accepted part of many organizations' information security tool suite. We are concerned that organizations are counting on these tools to solve a class of problems before they fully understand them. As a result, the solutions are likely to be inadequate or incorrect. Over-reliance on ID technologies can create a false sense of confidence about the degree to which tools are detecting intrusions against an organization's critical assets.

Both through our own experience and in discussion with technology experts and market analysts, we have observed that the current market condition of commercial ID tools and technologies exhibits a growing "bandwagon" effect. Each organization is comparing what they are doing with others in their peer group or market segment. If an organization views itself as taking security protection actions (such as deploying an IDS) that are equal to or slightly better than an organization that it considers its peer, that is good enough. At the decision-making level, there appears to be little or no regard for what ID systems can actually do. Nor is there an appreciation for the tasks that ID systems should not (or cannot) be relied upon to perform. Management's priority appears to be to ensure that they can demonstrate that they have exercised a standard of due care in the event of any legal action. We believe that the vendor community is marketing to this condition through the product claims they make.

It remains to be seen whether or not intrusion detection technology can live up to the promise of accurately identifying attacks. The current generation of commercial ID systems uses a limited set of techniques to detect attacks. Attackers are rapidly improving their abilities to penetrate networks successfully — for example by developing ways to defeat ID systems themselves. Challenges to today's ID systems include

- increases in the types of intruder goals, intruder abilities, tool sophistication, and diversity, as well as the use of more complex, subtle, and new attack scenarios
- the use of encrypted messages to transport malicious information
- the need to interoperate and correlate data across infrastructure environments with diverse technologies and policies
- ever increasing network traffic
- the lack of widely accepted ID terminology and conceptual structures
- volatility in the ID marketplace which makes the purchase and maintenance of ID systems difficult
- risks inherent in taking inappropriate automated response actions
- attacks on the ID systems themselves
- unacceptably high levels of false positives and false negatives, making it difficult to determine true positives
- the lack of objective ID system evaluation and test information
- the fact that most computing infrastructures are not designed to operate securely
- limited network traffic visibility resulting from switched local area networks. Faster networks preclude effective real-time analysis of all traffic on large pipes.

ID systems can provide useful, reliable results in specific situations and configurations. These include monitoring an organization's firewall policy to ensure it is implemented correctly, monitoring unpatched machines for specific vulnerabilities, and monitoring specific network services.

The key deployment consideration is to focus the IDS sensing and analysis activities on the most critical subnets and hosts so that a trained analyst can interpret and act on the data these activities produce to safeguard the most important assets.

This report presents recommendations for ID sponsors, users, vendors, and researchers. For sponsors, we recommend

- supporting ongoing, comprehensive testing of commercial ID systems and making test results publicly available
- emphasizing research funding directed towards reducing false alarms

For users, we suggest

- implementing a security architecture that reflects a defense-in-depth or layered approach to protecting an organization's assets, whether or not the organization chooses to deploy an IDS
- developing clear, concise IDS requirements based on security policy and organizational needs
- configuring the IDS to maximize performance. This includes selective deployment to monitor critical assets as well as signature tuning to prevent excessive false alarms.

We recommend that vendors

- support initiatives to create open source signatures
- move towards the distribution model used by the anti-virus community
- spend more time and resources testing signatures and making results public
- provide measures that represent the level of confidence a user should place in an ID system's ability to report an intrusion by type of signature or attack
- integrate human analysis as part of event diagnosis
- integrate available data sources more effectively to include information from different sensors and from different ID systems
- expand options for capturing forensic evidence

- increase efforts to detect malicious code (email attachments, Java, ActiveX)
- increase interaction with the research community

We believe that the research community can benefit the ID field by

- emphasizing the integration of diverse sources of available data to reduce false alarms
- providing credible, defensible test data to support test and evaluation of ID systems
- providing a taxonomy of vulnerabilities, i.e., a taxonomy that takes a victim rather than an intruder perspective
- developing approaches for defending against sophisticated attacks such as denial of service, insertion, evasion, and distributed, coordinated attacks
- developing approaches that integrate human analysis as part of event diagnosis
- developing approaches that support better detection of malicious code
- increasing interaction with the vendor community

This report does not emphasize current Department of Defense (DoD), Air Force (AF), and Defense Information Assurance Program initiatives in intrusion detection systems and technologies. Many of these efforts are specific to the DoD and involve proprietary products, systems, and documentation. In addition, we believe that the DoD and AF are well informed on the ID-related initiatives they are sponsoring. They are supported by other federally funded research and development centers (FFRDCs) (such as MITRE) in this area. Thanks to the Air Force Information Warfare Center (AFIWC), we have included a brief description of Government Off-the-Shelf (GOTS) ID efforts in Section 2.1.4. Our general approach was to analyze publicly available sources that could be of potential use to the DoD and to the general consumer, vendor, and research communities.

Preface

Because most deployed computer systems are vulnerable to an ever increasing threat of attack, intrusion detection (ID) is a rapidly developing field. Intrusion detection is an important technology business sector as well as an active area of research. Vendors make many claims for their products in the commercial marketplace so separating hype from reality can be a major challenge.

A goal of this report is to provide an unbiased assessment of publicly available ID technology. We hope this will help those who purchase and use ID technology to gain a realistic understanding of its capabilities and limitations. The report raises issues that we believe are important for ID system developers to address as they formulate product strategies. The report also points out relevant issues for the research community as they formulate research directions and allocate funds.

This report does not emphasize current Department of Defense (DoD), Air Force (AF), and Defense Information Assurance Program initiatives in intrusion detection systems and technologies. Many of these efforts are specific to the DoD and involve proprietary products, systems, and documentation. In addition, we believe that the DoD and AF are well informed on the ID-related initiatives they are sponsoring. They are supported by other federally funded research and development centers (FFRDCs) (such as MITRE) in this area. Thanks to the Air Force Information Warfare Center (AFIWC), we have included a brief description of Government Off-the-Shelf (GOTS) ID efforts in Section 2.1.4. Our general approach was to analyze publicly available sources that could be of potential use to the DoD and to the general consumer, vendor, and research communities.

Section 1 of the report provides an overview of ID technology from the perspective of the CERT® Coordination Center (CERT/CC).[1] The rapid growth in intrusion activity is fueling an increasing need for ID technology. This section provides context by citing examples that demonstrate how vulnerable networks and systems have become. It is followed by a review of the elements of attacks from the perspective of the attacker and of the victim. To convey how challenging it is to detect intruders, the dimensions of ID technology are characterized. Finally, this section reviews some of the challenges that confront the field of intrusion detection.

---

1. CERT and CERT Coordination Center are registered in the U.S. Patent and Trademark Office. In response to the attack of the Morris worm in 1988, the Defense Advanced Research Projects Agency (DARPA) decided to create the CERT® Coordination Center (CERT/CC) at the Software Engineering Institute (SEI). The SEI was charged with establishing a capability to quickly and effectively coordinate communication among experts during security emergencies in order to prevent future incidents and building awareness of security issues across the Internet community. Since its inception in 1988, the CERT/CC has responded to more than 20,000 security incidents that have affected over 400,000 sites in the Department of Defense (DoD), other federal agencies, and the private sector. For more information, refer to the CERT/CC Web site at http://www.cert.org.

Section 2 provides an in-depth look at the current state of ID technology. The section starts with a review of research, commercial, and publicly available tools, and then examines the rate at which industry is adopting commercial products.

We describe some informal experiments we performed with a variety of commercial and research ID tools. Finally, we present what we believe are the some benefits and shortcomings of the current generation of ID tools.

Section 3 reviews a wide range of issues that need to be confronted if ID systems are to become an effective technology and suggests some solutions. Much of the vendor literature conveys a perception that if one installs an IDS, one no longer has to worry about undetected intrusions. Unfortunately, this is not the case. The issues are broad-ranging and include external pressures from attackers, human factors, and limitations in the current technology. While technology may solve part of the intrusion detection problem, it is likely to be ineffective unless it fits within the organization's business objectives and operations.

Section 4 suggests practices that an organization should adopt if they want to derive the greatest benefit from an IDS.

Section 5 provides recommendations for the intrusion detection sponsor, user, vendor, and research communities.

The appendices provide supporting information in several areas.

Appendix A defines terms as they are used in this report. Terminology is not applied consistently given the immaturity of the ID field so having a set of definitions is important.

Appendix B provides a list of references.

Appendix C defines acronyms used in this report.

Appendix D contains a review of selected ID technology literature, providing supporting detail for Section 2.1.

Appendix E identifies organizations and standards relevant to intrusion detection.

Appendix F provides a candidate set of criteria that can be used in selecting an intrusion detection system.

All sources (Appendix D) and related efforts (Appendix E) reviewed in preparation of this report are current as of January, 2000.

It is important to note that the scope of this report is also defined by what it does *not* address:

- a detailed technical explanation of intrusion detection principles and how the technology works
- operational issues associated with installing, deploying, and managing an IDS, other than as briefly described in Section 4
- threat management, including
  - the incorporation of IDS management within CSIRTs (computer security incident response teams)
  - the role of IDS in threat management, such as defining alarm severity, monitoring, alerting, and policy-based actions
  - the role of the IDS administrator (such as converting IDS logs into forensic evidence)
  - the development of event response procedures
  - the recommendation of enterprise-wide policies based on threats
- physical security including physical intrusion detection and intrusion detection systems

This report contains many Web references. The intrusion detection field changes rapidly and much information is posted first (and often only) on the Web. Many of these references either become out of date, are modified, or disappear altogether from the original site. During the development of this report, this was a problem. Consequently, we have downloaded a majority of the references into an electronic repository that we can access in the event Web pages are subsequently modified or removed from their original location. This is a somewhat unusual approach but, given the increasing dominance of the Web, we believe that it will become more prevalent.

As a cautionary note, we strongly urge you not to rely on Web references cited in this report (or any other report that is more than three months old) for detailed IDS product information unless you verify that the data is correct. This caution is extended to reports on details about attacks and how these attacks manifest themselves through various monitoring mechanisms.

1 Intrusion Detection — What Is It and Why Is It Needed?

1.1 The Seriousness of Cyber Attacks

Attacks on the nation's computer infrastructures are becoming an increasingly serious problem. Even though the problem is ubiquitous, government agencies are particularly appealing targets and they tend to be more willing to reveal such events than commercial organizations. This is demonstrated by the cases cited below. While statistics on the growth of attacks provide a more solid basis for justifying the need for intrusion detection (ID), case histories can often be more persuasive.

October 7, 1999: *Hackers apparently working from Russia have systematically broken into Defense Department computers for more than a year and took vast amounts of unclassified but nonetheless sensitive information, U.S. officials said Wednesday. Besides penetrating the Pentagon's defenses, the hackers have raided unclassified computer networks at Energy Department nuclear weapons and research labs, at the National Aeronautics and Space Administration and at many university research facilities and defense contractors, officials said.* [N9]

October 7, 1999: *At NASA, the attacks are "massive, really very massive," and "very, very surreptitious," NASA Inspector General Roberta Gross said in an interview. "It's difficult to tell what the damage is," Gross said. "They weren't shutting down systems. They were taking file listings, looking to see what's in people's directories." Gross said the intruders also installed "parking tools that they can use to get back in later." Such electronic "trap doors" may be used to evade detection devices and to secretly regain access.* [N9]

June 1, 1999: *After NATO jets hit the Chinese Embassy in Belgrade in May, hackers from China attacked a handful of U.S. government sites, including one maintained by the Energy Department. In an unrelated incident, the official White House site was shut down briefly because of an attempt to tamper with it by unidentified hackers, officials said.* [N1]

May 21, 1999: *"We successfully penetrated several mission-critical systems, including one responsible for calculating detailed positioning data for Earth-orbiting spacecraft and another that processes and distributes the scientific data received from these spacecraft," the General Accounting Office (GAO) said...Having gained access to these systems, the report*

said, "We could have disrupted NASA's ongoing command and control operations and stolen, modified, or destroyed system software and data." [N3]

May 11, 1999: *The White House Web site was shut down today to determine whether hackers who tried to tamper with the site managed to do so. White House spokesman Barry Toiv said the site was shut down for 24 hours beginning late yesterday...MSNBC reported that there have been a series of politically motivated raids on government sites, undertaken in protest of last week's NATO bombing of the Chinese embassy in Belgrade, Yugoslavia. Unnamed government sources said the departments of Energy, Interior, and Labor as well as the U.S. Information Agency recently have been hit.* [N4]

April 6, 1999: *The nation's three nuclear weapons labs have shut down their classified computer systems for at least a week to beef up network security. Three preeminent Energy Department facilities halted operations Friday on all computers that handle secret information, in response to an unfavorable information security rating in a DOE audit of last year, according to Los Alamos National Laboratory spokesman Jim Danneskiold. The other two labs affected by the shutdown are Lawrence Livermore National Laboratory and Sandia National Laboratories...All three facilities will undertake several initiatives to improve security, including conducting computer security and threat awareness training; devising stricter access policies and tougher enforcement; implementing more rigorous procedures for transferring information from classified to unclassified computers; and establishing new intrusion detection measures.* [N5]

March 31, 1999: *NATO spokesman Jamie Shea said service on NATO's home page had been "erratic to say the least" since March 28, the fifth day of the alliance's bombing campaign against Yugoslavia. "It seems that we have been dealing with some hackers in Belgrade, who have hacked into our Web site," Shea told a news conference at NATO headquarters in Brussels. "At the same time, our email system has also been saturated by one individual who is currently sending us 2,000 emails a day. We are dealing with macro viruses from Yugoslavia in our email system," he said. A senior NATO diplomat said it was clear how well-organized and prepared Belgrade's offensive was: "It ranges all the way from organized ethnic cleansing to messing up our Web site."* [N6]

March 5, 1999: *The Pentagon today confirmed that attacks against U.S. military computers over the past few months are under special investigation by law enforcement and intelligence authorities. Deputy Defense secretary John Hamre briefed the House Armed Services Committee on the matter in a classified meeting February 23, according to the House Armed Services Committee. He warned legislators that the attackers were not merely individual hackers, and said part of the problem may stem from the cooperation of insiders within the U.S. military staff.... Hamre told the committee that the Pentagon detects between 80 and 100 hacker "events" every day. The Pentagon must investigate approximately one in ten of these.... One security expert said that while attacks from Russian and other foreign nations was*

nothing new, the new breed of hacks posed grave threats in their sophistication. "There is a steadily increasing number of these attacks," said Alan Paller, director of research for The SANS Institute. "And there are more of these that have three characteristics that set them apart." The first of these is that attacks are coming simultaneously from multiple, coordinated sites. The second is that the attacks are coming with more stealth, escaping the detection of intrusion monitoring systems by limiting the number of "pings," or connections. "These are coming in just under the detection threshold, at one every hour, or every three days," said Paller. "They're coming from patient people, who are usually more professional than children." [N7]

September, 1998: *Hackers are banding together across the globe to mount low-visibility attacks in an effort to sneak under the radar of security specialists and intrusion detection software, a U.S. Navy network security team said today. Coordinated attacks from up to 15 different locations on several continents have been detected, and Navy experts believe that the attackers garner information by probing Navy Web sites and then share it among themselves. "These new patterns are really hard to decipher — you need expert forensics to get the smoking gun," said Stephen Northcutt, head of the Shadow intrusion detection team at the Naval Surface Warfare Center. "To know what's really happening will require law enforcement to get hold of the hackers' code so we can disassemble it."* [N8]

1.2 The Rapidly Growing Threat

The press releases in Section 1.1 reflect the serious and sophisticated nature of recent cyber-attacks. This is compounded by the fact that, over the past 12 years, the growth of incidents on the Internet has reflected the growth of the Internet itself. Figure 1-1 illustrates this growth by plotting the number of incidents reported to CERT/CC over those years. E-commerce can only exacerbate the upward trend in incidents. While in previous years, external attacks tended to originate from those interested in exploring the Internet for its own sake and testing their skills, there is an increasing trend towards intrusions motivated by financial, political, and military objectives. From a recent survey of 91 respondents [S33], there was an average loss of $256,044 per respondent's organization. While the survey indicated that internal breaches were still of greater concern, external attacks were increasing at an "alarming rate." The survey stated that "...the number of companies experiencing [penetration attacks] doubled from about 12 percent of all respondents in 1998 to 23 percent this year." Thus the stakes are being raised.

In the 1980s, intruders were the system experts (Figure 1-2). They had a high level of expertise and personally constructed methods for breaking into systems. Use of automated tools and exploit scripts was the exception rather than the rule. Today, absolutely anyone can attack a network — due to the widespread and easy availability of intrusion tools and exploit scripts that duplicate known methods of attack.

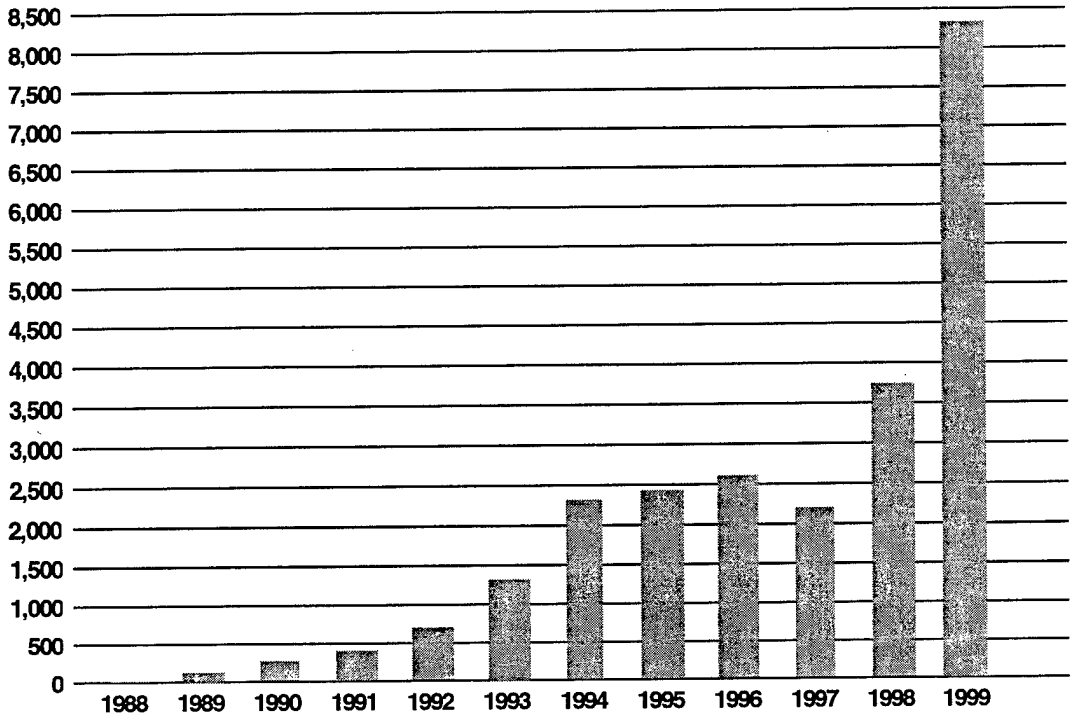


FIGURE 1-1: GROWTH IN NUMBER OF INCIDENTS HANDLED BY THE CERT/CC®

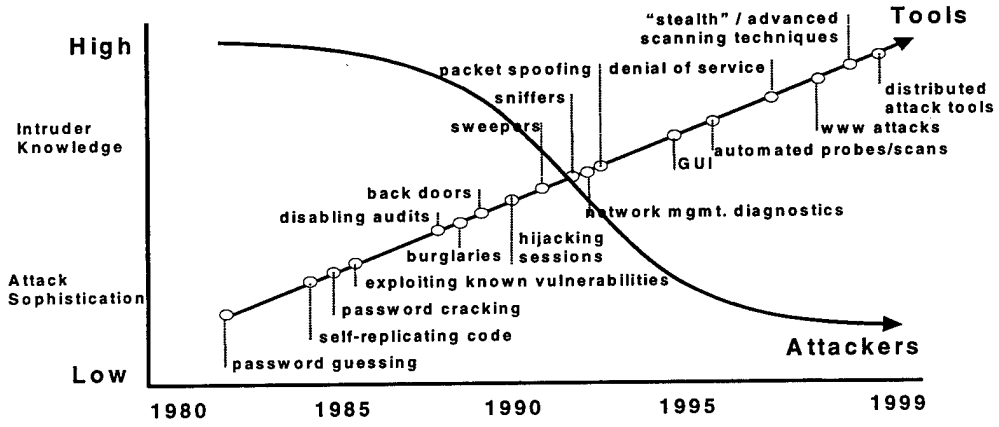


FIGURE 1-2: ATTACK SOPHISTICATION VS. INTRUDER TECHNICAL KNOWLEDGE

While experienced intruders are getting smarter, as demonstrated by the increased sophistication in the types of attacks, the knowledge required on the part of novice intruders to copy and launch known methods of attack is decreasing.

In the early/mid 1980s, intruders manually entering commands on their personal computer could access tens to hundreds of systems; today, intruders use automated tools to access thousands to tens of thousands of systems.[1] In the 1980s, it was relatively straightforward to determine if an intruder had broken into your systems and to determine their actions. Today, intrusions, and the damage they cause, can occur in a matter of seconds. Intruders are able to totally hide their presence by, for example, disabling commonly used services and reinstalling their own versions, and by erasing their tracks in audit and log files. In the 1980s and early 1990s, denial-of-service attacks were infrequent and not considered serious. Today, for organizations that conduct business electronically, such as online stock brokers and traders, a successful denial of service attack can put them out of business. As shown in Figure 1-1, from 1997 to 1998, we saw a 75 percent increase in the number of incidents reported to the CERT/CC. In 1999, the number of incidents increased by over 120 percent from 1998.

There are many reasons for the growing number and severity of attacks, including increased connectivity and complexity, increased availability of vulnerability information and attack scripts via the Internet, and dependence on distributed network services. As indicated by Donn Parker, the very nature of computer crime is that it is unpredictable, so you can't use previous threats or attacks as a metric to prepare for future threats or attacks — the basis for all of today's signature-based ID products [B91].

1.3 Attacker and Victim Perspectives on Intrusion

Attacks and intrusions can be viewed from a number of perspectives. The most common are those of the intruder and the victim. Each perspective brings with it distinct criteria for judging the success of the attack. Typically, we say that an intrusion has taken place when an attack is considered successful from the victim's perspective, i.e., the victim has experienced some loss or consequence. A successful attack is enabled by the presence of a vulnerability in the victim's system that is exploited by an intruder with an objective. We use the term intrusion to mean a successful attack.

An attack is unsuccessful from the perspective of the intruder if none of their objectives are fulfilled; whereas, a victim perceives an attack as unsuccessful if there are no consequences that result from the attack. Unsuccessful attacks from the perspective of an intruder may still have one or more consequences for a victim.

---

1. Based on CERT/CC experience.

The intrusion process begins when an intruder takes steps to fulfill an objective. An essential component of an intrusion is taking advantage of one more vulnerabilities by using tools and exploit scripts.

The vulnerabilities exploited in this process can range from a flaw in a piece of software, such as a buffer overflow that can be exploited to elevate privileges, to a flaw in an organizational structure that allows a social engineering attack to obtain sensitive information or passwords to accounts. The intrusion process ends when some or all objectives of the intruder are realized or the intruder gives up.

Attacks can involve one or more attackers and one or more victims. Because multiple perspectives are involved in a single attack, defining what constitutes an attack is difficult. Is an attack an action taken by an adversary or is it the manifestation of that action as observed by the victim? Consider the example of the smurf attack [B92] where the attacker convinces a third party to perform an undesirable action against the intended victim. Is the attack the attacker pressing the enter key on the shell command to execute smurf? Is the attack the series of packets sent to the third party? Or is the series of packets observed at the victim site the attack? Or do all of these events fit together to define an attack?

Some example components of an attack from the perspective of an intruder are

- objective
- exploit scripts
- vulnerabilities in target system
- risk of carrying out an intrusion
- damage caused or consequences to victim

Components of an attack fall into either a known or an unknown category. Because attacks contain multiple components, it is possible that a single intrusion contains both. The concept of known and unknown varies from different perspectives of intrusion as well. An intruder does not necessarily know the consequences of an intrusion for a victim site or all of the hosts that were affected by an intrusion. Similarly, a victim does not necessarily know the objective of an intruder, the exploit scripts that are used, the vulnerabilities that are exploited, or the identity of the intruder.

Some example components of an attack from the perspective of a victim are

- What happened?
- Who is affected?
- How are they affected? (consequences)

- Who is the intruder?
- Where did the intrusion originate?
- When did the intrusion occur?
- How did the intrusion happen?
- Why did the intrusion happen?

The goal of intrusion detection is to positively identify all true attacks and negatively identify all non-attacks. The motivation for using intrusion detection technology may vary for different sites. Some may be interested in law enforcement including the tracking, tracing, and prosecution of intruders. Some may use intrusion detection as a mechanism for protecting computing resources, while others may be more interested in identifying and correcting vulnerabilities.

1.4 Dimensions of Intrusion Detection

Just as attacks can be viewed in different ways, so can the process of detecting them. Intrusion detection can result from the observation of an attack in progress or from recognizing the results of an intrusion after the fact. This section summarizes several characteristics of intrusion detection.

As the terms are used below, there is a discrepancy; the term intrusion is used to connote a successful attack and it is also used in the phrase "intrusion detection system" describing a system designed to detect attacks regardless of their success. To be semantically correct and consistent, we should use the phrase "attack detection system" to represent such a system; however, we continue to use the phrase "intrusion detection system" with the understanding that unsuccessful attacks are also represented.

1.4.1 Terminology

Intrusion detection is a young field, and many terms are not used consistently. As discussed above, there is even disagreement about what is meant by "intrusion" and "attack." There are multiple terms used to represent various methods of detecting intrusions. This section clarifies the meaning of some important ID concepts as they are used throughout this report. A more complete set of definitions can be found in Appendix A.

- Analysis approaches
  An analysis approach is a method used by an IDS to determine whether or not an intrusion has occurred. There are two major categories of analysis approaches:
  - Attack signature detection (sometimes called "misuse detection") identifies patterns corresponding to known attacks.

This includes passive protocol analysis which is the use of sniffers in promiscuous mode. It also includes signature analysis which is the interpretation of a series of packets (or a piece of data contained in those packets) that are determined, in advance, to represent a known pattern of attack [B26-b].

The attack signature may also be manifest in audit records, logs, or in changes in the compromised system.

- Anomaly detection identifies any unacceptable deviation from expected behavior. Expected behavior is defined, in advance, by a manually developed profile or by an automatically developed profile. An automatically developed profile is created by software that collects and processes characteristics of system behavior over time and forms a statistically valid sample of such behavior. Note that unexpected behavior is not necessarily an attack; it may represent new, legitimate behavior that needs to be added to the category of expected behavior. A comparison between these approaches is given in Section 1.4.4.

- Attack
  An action conducted by one adversary, the intruder, against another adversary, the victim. The intruder carries out an attack with a specific objective in mind. From the perspective of an administrator responsible for maintaining a system, an attack is a set of one or more events that may have one or more security consequences. From the perspective of an intruder, an attack is a mechanism to fulfill an objective.

- Exploit
  The process of using a vulnerability to violate a security policy. A tool or defined method that could be used to violate a security policy is often referred to as an exploit script.

- False negative
  An event that the IDS fails to identify as an intrusion when one has in fact occurred [B26-b].

- False positive
  An event, incorrectly identified by the IDS as being an intrusion when none has occurred [B26-b].

- Incident
  A collection of data representing one or more related attacks. Attacks may be related by attacker, type of attack, objectives, sites, or timing.

- Intruder
  The person who carries out an attack. Attacker is a common synonym for intruder. The words attacker and intruder apply only after an attack has occurred. A potential intruder may be referred to as an adversary. Since the label of intruder is assigned by the victim of the intrusion and is therefore contingent on the victim's definition of encroachment, there can be no ubiquitous categorization of actions as being intrusive or not.

- Intrusion
  A common synonym for the word "attack"; more precisely, a successful attack. In this report, we often use the term intrusion to include attack, because the subject of the report is intrusion detection systems.

- Vulnerability
  A feature or a combination of features of a system that allows an adversary to place the system in a state that is contrary to the desires of the people responsible for the system and increases the probability or magnitude of undesirable behavior in or of the system.

1.4.2 ID System Components

The functionality of an IDS can be logically distributed into three components: sensors, analyzers, and a user interface.

- Sensors
  Sensors are responsible for collecting data. The input for a sensor may be any part of a system that could contain evidence of an intrusion. Example types of input to a sensor are network packets, log files, and system call traces. Sensors collect and forward this information to the analyzer.

- Analyzers
  Analyzers receive input from one or more sensors or from other analyzers. The analyzer is responsible for determining if an intrusion has occurred. The output of this component is an indication that an intrusion has occurred. The output may include evidence supporting the conclusion that an intrusion occurred. The analyzer may provide guidance about what actions to take as a result of the intrusion.

- User interface
  The user interface to an IDS enables a user to view output from the system or control the behavior of the system. In some systems, the user interface may equate to a "manager," "director," or "console" component.

In addition to these three essential components, an IDS may be supported by a "honeypot," i.e., a system designed and configured to be visible to an intruder and to appear to have known vulnerabilities. A honeypot provides an environment and additional information that can be used to support intrusion analysis. The honeypot serves as a sensor for an IDS by waiting for intruders to attack the apparently vulnerable system. Having a honeypot serve as a sensor provides indications and warnings of an attack. Honeypots have the ability to detect intrusions in a controlled environment and preserve a known state.

1.4.3 Integrating Detection and Response

Intrusion detection and response have traditionally been thought of as two separate processes; however, the line between them is beginning to blur. As ID systems continue to evolve and improve, they are beginning to incorporate limited capabilities to respond to intrusions.

Typical responses to intrusions may include dropping suspicious traffic at the firewall, denying user access to resources as they exhibit anomalous behavior, or reporting the activity to other hosts or sites involved in the attack.

In Section 1.4.4, we describe a hierarchical model for intrusion detection systems, and explain that output from these systems tends to travel from the lower levels to the higher. Response data on the other hand may travel in either direction. For example, a network-based IDS may provide host-level response, such as modifying configuration files on particular hosts. Response may include updating configurations for other IDS components meaning that a response of one IDS or component could have an effect on the behavior of another IDS or component. For these reasons, detection and response systems are beginning to merge.

1.4.4 ID Systems "Hierarchy"

Although every IDS can be conceptually viewed as having a sensor, an analyzer, and a user interface, the types of data examined and the types of data generated by a particular IDS may vary significantly. ID systems can be classified into one of the following categories based on the types of data they examine:

- Application
  An application-based IDS examines the behavior of an application program, generally in the form of log files.
- Host
  A host-based IDS examines data such as log files, process accounting information, user behavior, or outputs from application-based ID systems operating on the host.
- Network
  A network IDS examines network traffic. It may have access to outputs from host-based and application-based ID systems operating within the monitored network environment.
- Multi-network/infrastructure
  A multi-network IDS generally takes the form of an incident response team (IRT), where the input of the system comes from "sites" within their constituency. A site in this case is an entity that lies within an administrative domain. Data communicated to this type of IDS is generally from application, host, network, or other multi-network intrusion detection systems.

The categories of ID systems listed above can be thought of as a hierarchy, the top of the hierarchy being multi-network or infrastructure-based ID systems and the bottom being application-based. An IDS at any point in the hierarchy could receive data from any level lower in the hierarchy in addition to a sensor that may operate at the same level. Output from an IDS can be utilized by other ID systems at the same or higher levels in the hierarchy.

1.4.5 A Comparison of ID Analysis Methods

There are distinct analysis methods for detecting known and unknown attacks. As discussed above, we defined these as attack signature detection and anomaly detection. In this section, we describe the differences between these two methods by examining several attributes.

- Knowledge
  Detecting intrusions requires either knowledge of possible intrusions or knowledge of the known and expected behavior of a system. In order for an IDS with a signature-based method to detect all attacks, it requires prior knowledge of all possible attacks. The IDS must recognize either the details of an attack or the patterns at a more abstract level that characterize the class of an attack. An anomaly-based system must have full knowledge of the expected behavior of the system to detect all attacks. In reality, neither of these is possible; they represent ideals.

- Ease of configuration
  Another attribute to consider is ease of configuration. A signature-based system in general requires significantly less configuration effort than a system to detect anomalies since the latter requires much data collection, analysis, and updating. Some systems allow users to create their own signature files which can increase the complexity of establishing the desired configuration.

  Anomaly-based systems in general are more difficult to configure because a comprehensive definition of known and expected behavior for a system is required. This demands that the user discover, understand, represent, and maintain the expected behavior of their system. In many cases, automated support is provided but this takes time to develop and the data that is used must be unambiguous.

- Reported data
  Signature-based ID systems generally produce conclusions based on pattern matching. The output of a signature-based system can vary from an alert message indicating that a particular signature has occurred to one that also provides supporting data that is relevant to the signature's occurrence.

  The output of anomaly-based ID systems generally produce conclusions based on statistical correlations between actual and expected behaviors. Additionally, anomaly-based systems tend to produce more data since anything outside the realm of expected behavior is reported.

- Reporting accuracy
  Signatures that are not specific and anomaly profiles that are not adequately specified to describe expected behavior both result in ID systems that produce potentially large numbers of false positives and false negatives.

Depending on the environment within which an IDS is deployed, a combination of methods (signature and anomaly) for all types of ID systems (application, host, network, multi-network) may be required for the most effective solution. Signature-based ID systems are not able to detect all possible intrusions because of inherent detection limitations, constantly evolving attacks and exploits, new vulnerabilities, and use of new exploit scripts. Anomaly-based systems generally report a larger number of false positives as expected behavior changes.

An advantage of an anomaly-based IDS is the ability to detect novel attacks that can bypass signature-based systems. Such attacks can be analyzed by a person who can then define attack signatures. Using the combination of signature- and anomaly-based methods provides the capability to detect a larger variety of attacks and keep the signature-based system up to date.

1.5 Operational Challenges with Intrusion Detection Systems

Implementing intrusion detection systems on networks and hosts requires a broad understanding of computer security. The complexity of information technology infrastructures is increasing beyond any one person's ability to understand them, let alone administer them in a way that is operationally secure. Vendors are rapidly releasing new ID systems and aggressively competing for market share in an expanding market. Many products started out as point solutions. However, in response to consumers' inability to fully understand and use many ID systems, vendors are attempting to integrate approaches to solve a broader range of computer security problems. Evaluating ID systems is non-trivial and there is a lack of credible, comprehensive product evaluation information. Hiring and retaining personnel to competently administer security in general and intrusion detection in particular are increasing challenges. All of this rapid change makes it very difficult for an organization to implement an effective, long-term security strategy.

1.5.1 Growth in the Number and Claims of ID Products

Intrusion detection is an important and rapidly growing security technology market. International Data Corporation (IDC) reports revenues for these products increased 135 percent to $136 million in 1998 — and the growth is just getting started.

In 1999, the market was projected to grow almost 100 percent, and by 2003, it will approach $980 million [B93]. This market growth is driven by reports of steadily increasing number of computer security breaches; a 22 percent rise from 1996 to 1998, with $136 million in associated losses [B94]. Intrusion detection is considered by many to be the logical complement to network firewalls, extending the security management capabilities of system administrators to include security audit, monitoring, attack recognition, and response [B23].

Clearly, in this type of fast-paced, growing market, security product vendors are eager to capture market share and make claims that will support their efforts. However, regardless of vendor claims, ID systems do not have the capability to look at every possible security event. The event could have happened on a different network, the IDS itself could have been compromised, or the IDS might have reached its maximum bandwidth capacity and dropped further network traffic [B76]. Most commercial products have their own proprietary protocol for communications between the sensor detecting the event of interest and the analysis function that interprets the significance of the event — which makes it virtually impossible to correlate information from multiple ID systems or monitoring tools.

ID systems themselves are logical targets for attack [B26-b]. Smart intruders who realize that an IDS has been deployed on a network they are attacking will likely attack the IDS first, disabling it or forcing it to provide false information (distracting security personnel from the actual attack in progress, or framing someone else for the attack). In addition, many commercial and research ID tools have carried forward original design assumptions resulting in security weaknesses such as sending log files without encrypting them, absence of access control, and not performing integrity checks of ID system files.

1.5.2 Difficulty with Evaluating ID Technologies

It is extremely difficult to identify and evaluate the processes, procedures, tools, software, hardware, and databases that comprise the range of ID technologies. There is no industry standard against which to compare such systems because the technology is too new. The commercial ID new product cycle is very fast, based on the pace of the growing market. Northcutt [B76] recommends that you only use publications that are updated at least monthly as ID product buyer's guides. Even after an organization has identified a list of candidate ID system solutions, the evaluation process will be quite complex if it is to provide the answers required to make an informed decision.

Marketing literature rarely describes how well a given IDS finds intruders and how much work is required to use and maintain that system in a fully functioning network with significant daily traffic. IDS vendors can specify which prototypical attacks can be found by their systems, but without access to the normal traffic generated by day-to-day work, they cannot describe how well their systems detect real attacks while passing background traffic and avoiding false alarms.

This information is critical: every declared intrusion requires time to review, regardless of whether it is a correct detection for which a real intrusion occurred, or whether it is merely a false alarm [B95].

Evaluating ID system capabilities requires test data; either network traffic (for network-based approaches) or profiles (of systems, processes, file use, and user behavior for host-based approaches). If you choose to do this yourself as part of the evaluation process, setting up the networks, operating environments, traffic samples (e.g., using live traffic or simulated "bad" traffic), and other supporting data is non-trivial and requires a significant investment of resources and time. Determining the intrusion detection approach employed by each product and the ways in which intrusions are detected is also non-trivial. These topics are described in more detail in Section 2.3 and 3.5.1.

Some organizations and standards groups are attempting to address many of the issues surrounding the selection and use of ID technologies. Several of these efforts are described in Appendix A.

1.5.3 Maintaining Necessary Knowledge

Given the constantly changing landscape of attacks and intrusions, you need to maintain several types of information (based on the IDS analysis approach) to ensure that your IDS continues to detect suspicious events. According to Amoroso [B89], this information includes

- profiles of normal and abnormal user, system, and process behavior
- strings that denote suspicious traffic patterns, including signatures of known attacks and intrusions
- information used to initiate response actions to various anomalies and attacks

Some of this information is likely maintained by the IDS vendor but not necessarily all. Staff responsible for the IDS should obtain the information in a secure manner and arrange for installed ID systems to be regularly updated (similar to operating system patches or new viruses being loaded into virus detection software). Any information not provided by the vendor must be maintained and applied when needed by technical staff.

1.5.4 Lack of Qualified Technical Staff

Technology alone is not enough to maintain network security. An organization needs qualified technical staff to evaluate, select, install, operate, and maintain ID technologies. In today's market, there is a decreasing availability of the qualified intrusion analysts and system/ network administrators who are knowledgeable about computer security.

According to Northcutt [B76], having an ID product do your "thinking/analysis" for you is a natural response to the lack of skilled technical people, particularly those with security skills. However, there are many attacks and probes occurring every day that are not canned "script kiddie" exploits. Only trained analysts with expert-class tools are going to be able to detect and analyze these.

As Amoroso indicates [B89], nearly all *reported* incidents in which an intruder has been caught in real time have involved manual intrusion detection methods used by attentive security experts. Furthermore, these incidents have involved locally developed ID tools and traps rather than commercial systems [B89]. Automation of the entire ID process is unlikely in the near future.

In the face of this reality, many organizations are choosing to outsource the ID operations, an option that comes with its own issues and risks (see Section 4.6 for further information).

1.5.5 Intrusion Detection as a Component of Defense-in-Depth

Intrusion detection is needed because firewalls cannot provide complete protection against intrusion. Experience teaches us never to rely on a single defensive line or technique. A firewall serves as an effective noise filter, stopping many attacks before they can enter an organization's networks. However, firewalls are vulnerable to errors in configuration and ambiguous or undefined security policies. They are generally unable to protect against malicious mobile code, insider attacks, and unsecured modems. Firewalls rely on the existence of a central point through which traffic flows when the growing trend is towards geographically distributed networks with inside and outside users traversing the same subnets and, therefore, the absence of central points for firewall monitoring purposes. On internal networks, routers or switches can be configured to watch for signs of intrusion and take appropriate action based on what they detect [B76].

Implementing multiple layers of protection as part of an overall security architecture (such as firewalls, access control and authentication mechanisms, monitoring tools, vulnerability scanning tools, ID systems, security training) makes penetration by external intruders more difficult while making intrusion prevention and detection somewhat easier. See Section 4.5 for more details on this point.

2 What Is the Current State of Intrusion Detection Technologies?

This section covers a range of topics dealing with current ID technology and practice to illustrate where ID systems stand today. We start with a review of technology, looking at currently used tools in the research, commercial, and public domains. We then look at market conditions, summarizing several papers and surveys that describe the current market and where it is headed. We conclude by discussing some experiences with representative ID products that indicate why current ID systems are not the only solution to fix all security problems.

2.1 Survey of ID Technology

ID technology is immature and dynamic. Like the early auto manufacturers, new vendors appear, only to be absorbed by other vendors. The same is true with ID products (both commercial and research). Because of the rapid changes in the field, information such as lists, surveys, or reviews are quickly outdated. For example, a report by Staniford-Chen provides a summary of 42 ID-related products, but much has changed since its date of publication (winter 1997-1998).[1] Web-based lists (such as the one found in the SANS/NSA ID tools inventory) are easier to keep updated [B4].

As part of this effort, we conducted a wide-ranging review of ID literature. This review focuses mostly on materials accessible via the Web. Appendix B contains references to the articles reviewed in this section while Appendix D contains selected reviews.

Intrusion detection has been an active field of research for about two decades. This is exemplified by an influential paper, published in 1980, "Computer Security Threat Monitoring and Surveillance" by James Anderson [B34]. It was followed some years later (1987) by the seminal paper "An Intrusion Detection Model" by Dorothy Denning [S7]. Denning's paper provides a methodological framework that inspired many researchers and, in more recent times, laid the groundwork for commercial products.

1. In fact, the document and its URL were both revised while our report was being written. This change is reflected in the bibliography. A reference to the new document can be found in Appendix B [S12].

There are six main topic areas covered by the survey: ID surveys, taxonomies, testing and evaluation, research, commercial tools, and ID directions (all are further elaborated in Appendix D). The papers that are more relevant to this review are discussed more fully, while other papers are simply cited because they are useful resource materials. In all cases, the reviews are brief and are provided so that the reader can selectively target relevant papers. Each review follows the same structured table-based format.

This section presents some research and commercial products that are examples of available ID technology, as well as a few products available in the public domain.

2.1.1 Examples of Research Products

ID research performed in the early 1990s produced a number of new tools [S6]. However, many were developed by students to explore concepts, and after they moved on, the tools were not maintained. Nevertheless, these tools influenced the direction of subsequent research efforts and also of commercial ventures. Early efforts often focused on host-based solutions, but because of the explosive growth of networking, later efforts concentrated on network-based systems.

The tools reviewed here reflect a core of active research that has evolved from earlier efforts. The first two, EMERALD and NetSTAT, have matured a great deal and compliment each other's approaches. The third research tool discussed is Bro. It is unique in addressing the issue of network penetration through attempts to overload or confuse the ID system.

2.1.1.1 EMERALD

EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances) is the most recent research tool developed by SRI International. This line of tools has explored issues in intrusion detection associated with both deviations from normal user behavior (anomalies), and known intrusion patterns (signatures). SRI's pioneering work in intrusion detection began in 1983 when a multivariate statistical algorithm was developed to discriminate between different user behaviors [R1-d].

Somewhat later, the use of a signature analysis subsystem, based on the P-BEST [R33] expert system was investigated to support detection of suspicious activities. These research efforts were incorporated into SRI's early intrusion detection system, IDES [R1-b], a system that monitors activity on multiple hosts in real time.

Based on experiences with IDES, a re-achitected, production-oriented tool, NIDES [R1-c], was developed between 1992 and 1994. Like IDES, this tool is host-based, and uses the P-BEST production rule system.

However, it went further by adding a component called RESOLVER that fuses the results from the statistical and signature analysis components. The user interface in NIDES was also a significant improvement over the IDES user interface.

EMERALD builds on the earlier IDES/NIDES experiences but this time focuses on support for networks rather than for a collection of hosts. A major goal of EMERALD is to address issues associated with large, loosely coupled enterprise networks. Such environments are more difficult to monitor and analyze due to the distributed nature of the incoming information. EMERALD structures users into a federation of independently administered domains. Each domain provides a collection of network services, such as http or ftp, that may have different trust relationships with each other, and across which different security policies may apply. In this context, one centralized repository is likely to result in significant performance degradation, as is the centralized analysis of all the data. These issues motivated the work on EMERALD and the demonstration of "divide and conquer" techniques that it investigated.

The hierarchical approach provides three levels of analysis performed by a three-tiered system of monitors: service monitors, domain monitors, and enterprise monitors. These monitors have the same basic architecture: a set of profiler engines (for anomaly detection), signature engines (for signature analysis), and a resolver component that integrates the results generated from the engines.

Each module also contains a resource object that provides a configurable library of information to customize the module's components to the target application.This object can be re-used in multiple monitors within an EMERALD application. At the lowest level, service monitors support intrusion detection for individual components and network services within one domain, probing for or reading data (activity logs, events, etc.), and performing local signature and statistical analyses.

Domain monitors integrate information from the service monitors to provide a domain-wide view of intrusions, while the enterprise monitors perform inter-domain analysis to assess threats from a global perspective. Of interest at the enterprise level are such threats as worm-like attacks and inter-domain attacks on network services. Subscription-based communication channels allow different service monitors to communicate with each other, either by having information directed from one monitor to another ("pushed") or requested by one monitor from another ("pulled").

Prior work with NIDES demonstrated that statistical profiling techniques could be effective with either users or applications as targets. The monitoring of applications (e.g., anonymous FTP), was particularly effective since fewer application profiles were required. EMERALD generalizes the profiling technique by abstracting the notion of a profile, separating profile management from profile analysis.

With respect to signature analysis, service-layer signature engines monitor domain components to determine if abnormal activity is occurring through known exploit scripts. Signature engines in higher-level monitors distill this information to assess if a broader attack is occurring. In addition to integrating the results from the statistical and signature engines, the resolver component provides other functions. These include providing a subscription service that allows a third party tool to be integrated into the EMERALD environment, acting on reports generated by the statistical and signature engines, providing an interface to the monitor administrator, and initiating attack countermeasures, such as terminating processes.

EMERALD is a work in progress. It provides an example of the direction that future intrusion detection systems may take. As intruders become more sophisticated in their attacks, they will be increasingly likely to disperse the evidence of their work across networks, making it difficult to sense when a distributed/coordinated attack is occurring. In such situations, the ability to collect, assimilate, correlate, and analyze information emanating from diverse sources in real time becomes essential. The flexibility of EMERALD's scalable architecture, its ability to abstract functionality, its openness to the addition of external tools, and the prior experience (e.g., with NIDES) that is reflected in its functional components, makes EMERALD a forerunner of future ID tools. However, managing and maintaining the information base and building the system's infrastructure may require significant effort.

2.1.1.2 NetStat

NetSTAT is the latest in a line of "STAT" research tools produced by the University of California at Santa Barbara. The STAT activity, started in the early 1990s, explores the use of state-transition analysis in support of real-time intrusion detection [R4]. The approach is based on the premise that certain sequences of actions reflect unauthorized activity and indicate an intruder moving the system from an initial authorized state to a compromised state.

Most host-based intrusion detection systems in the anomaly category analyze evidence for intrusion in the computer's audit trail. However, in the STAT approach, the audit trail information is transformed through an "audit trail analyzer" that filters and abstracts the information gathered at the audit trail level. These abstractions, which are more suitable for analysis, portability, and human understanding, are called signatures and are central to the STAT approach. Signature actions move the system through the sequence of states, each state driving the system closer to a compromised configuration. Intrusion sequences are defined by state transitions that are captured in production system rule-sets.

The initial implementation of the method was a host-based, UNIX-based system called USTAT [R4]. USTAT was composed of

- *a preprocessor*
- *a knowledgebase (that included a fact base and rule base)*

- *an inference engine*
- *a decision engine*

The preprocessor filters and manipulates the data into a form that is audit-file independent. The rule base component of the knowledgebase stores the state-transition rules that indicate the predefined intrusion sequences, while the fact base stores the dynamically changing state of the system with respect to possible ongoing intrusions.

Given new information generated by the preprocessor together with the current system state as defined in the fact base, the inference engine identifies any significant state changes and updates the fact base. An update function then revises the fact base to reflect these changes. The inference engine also notifies a decision engine of possible security violations. The decision engine in turn either notifies the site security officer of the event or initiates action on its own. One advantage of the state-driven approach is that an attack may be recognized and acted on prior to reaching the compromised state.

This state-based approach uses an inference engine table to track each possible intrusion, and allows USTAT to identify a coordinated attack emanating from multiple sources. It can do this since attack sequences are defined, not by who is perpetrating the attack but by states of the system. Thus, if two attackers are relying on the same composite state of the system, each of their subsequent actions can be followed through a fork in the previous state transition sequence. This forking is implemented by duplicating rows in the inference engine table, each row representing different attack sequences.

NSTAT [R3] was the natural successor to USTAT. NSTAT focused on supporting a network of hosts in which, for example, files are shared. Thus actions on one host, such as mounting directories, can influence other machines on the network. Having one centralized detection system results in less performance impact on the local hosts and also allows for more informed intrusion analysis when a multi-host attack is being perpetrated. With NSTAT, the local hosts convert audit data into NSTAT format and merge the data steams into one.

The most recent of the tools, NetSTAT [R30], is currently under development and diverges from the prior host-based systems by addressing network intrusion. NetSTAT is composed of a set of probes that are responsible for detecting and evaluating intrusions in the sub-networks to which the probes are attached. Each probe is supported by a remotely configurable data filter, an inference engine, and a decision engine. These probes can act autonomously. However, different parts of the network may detect components of an intrusion (because of the differing locations and filters). If an intrusion component is detected, then an event can be forwarded to other interested probes that subscribe to that event in order to get a more complete understanding of the intrusion. Thus, intrusions that involve separate subnetworks can be identified.

The probes are supported by an analyzer, a stand-alone tool that supports the generation and management of probes. The analyzer is composed of a network fact base, a database of state-based intrusion scenarios, an analysis engine, and a configuration builder.

It determines which events should be monitored for, where they should be monitored, what network topology information is required, and what network state information is required to support the intrusion analysis. To perform these actions, the analysis engine uses information in the network fact base together with the scenario database to define attacks to which the network may be vulnerable. This information is passed to the configuration builder that in turn generates the probe configurations. These probe files consist of a filter, state-transition information, and the decision tables that allow the probe to execute.

2.1.1.3 Bro

Bro is a research tool being developed by the Lawrence Livermore National Laboratory. It is being built, in part, to explore issues related to the robustness of intrusion detection systems, i.e., assessing what characteristics make an ID system capable of resisting attacks against itself. The design goals for Bro [R31] include

- *high-load monitoring. The ability to handle high data transfer rates and traffic volumes without dropping packets is important. An intruder could use the mechanism of overloading the network with extraneous packets to flood the ID system. This could force the ID system to drop packets to which the network was vulnerable.*
- *real-time notification. This is needed to assure timely response to intruder threats.*
- *decoupling mechanism from policy. Separating the data filtering, event identification and policy reactions to the events results in a cleaner software design, easier implementation, and more straightforward maintenance.*
- *system extensibility. The large number of known attacks, together with the aggressive uncovering of new vulnerabilities, requires that Bro have the ability to rapidly add new attack scripts to its library.*
- *an ability to ward off attacks. Sophisticated attackers will likely probe for weaknesses in intrusion detection systems themselves.*

Bro has a three-level hierarchy of functions. At the lowest level, Bro uses libpcap, a utility to extract packets from the network. This decouples the main intrusion detection functionality of Bro from the networking details. This also allows a significant fraction of the packets entering the network to be rejected at a low level. Thus libpcap will capture all packets associated with the application protocols (e.g., finger, ftp, telnet) of which Bro is aware.

The next layer, the event layer, performs integrity checks on packet headers. If the header is ill-formed, an event identifying the problem is generated, and the header is discarded. A check is then performed to determine if the full contents of the packet should be recorded (usually if

the full packet was analyzed), if only the packet's header information should be recorded (usually if only the TCP flags were analyzed), or if nothing should be recorded (if no processing was done).

Events are generated from this process and placed on a queue to be interrogated by the policy script interpreter which resides in the third layer. The policy script interpreter is written in a customized Bro language that uses strong typing to provide explicit support for packet header content such as port and domain and other constructs to support networking concepts. The interpreter binds event values to the code for the event handler and then interprets the code. Executing the code may result in generating further events, logging real-time notifications, or recording data. To add new capability to Bro, one needs to identify the events associated with the protocols of the application, and write corresponding event handlers to extend the functionality of the policy script interpreter. The developer claims that this decoupling of events from their handlers improves Bro's extensibility.

At present Bro monitors four applications: finger, ftp, portmapper, and telnet. Adding new applications to Bro is, according the developer, "quite straightforward, a matter of deriving a C++ class to analyze each connection's traffic, and devising a set of events corresponding to the significant elements of the application" [R31]. Bro runs under several UNIX variants and is used as part of the lab's security system. As of 1998, Bro's operation had resulted in the filing of 85 CIAC and CERT/CC incident reports. Bro experiences no packet loss on a FDDI network carrying 25mbs traffic with analysis loads of peaking at about 200 packets/second.

2.1.2 Examples of Commercial Products

The commercial products described here represent only a small subset of those now on the market [see the references S12, B4, B3 for product lists], and the reviews of these products do not imply endorsement. Recent comparative evaluations can be found in references in Appendix B [S20, S21, S37]. The products discussed below represent a cross-section of approaches being employed in intrusion detection. In this context, we did not necessarily select the most dominant or fully functional products, but tried to pick those that reflected a cross-section of abilities spanning such dimensions as host-based vs. network-based and anomaly-based vs. signature-based. Also, there is a high degree of quality and detail in the product's literature. Currently most commercial systems are network-based. However, many commercial vendors are aggressively developing products that integrate host- and network-based approaches.

Unlike the research product literature, commercial product literature is generally weighted towards marketing. This can make it difficult to identify the functionality that the products actually support. For example, it requires close reading to identify a product's technical basis (is it host-based or network-based?). Virtually no commercial literature covers what are perhaps the most important topics to the prospective buyer: the frequency of false positives

(generating unnecessary alerts) and the frequency of false negatives (not identifying intrusions when they occur). While work is progressing on developing standardized test data [B29], accurate assessments of commercial tools are lacking.

2.1.2.1 CMDS™

The Computer Misuse Detection System (CMDS™) was originally developed by Science Applications International Corporation [C13], and is now maintained and marketed by ODS Networks Inc. [C11]. It is a host-based system that supervises a (potentially) tiered network of machines. It supports the anomaly (statistical) and misuse (signature) categories of detection and can also generate analysis reports depicting trends.

In the anomaly category, CMDS uses statistical analysis to identify patterns of behavior that deviate from normal user practice. The statistics are derived from such categories as

- login/logoff times
- applications executed
- numbers of files opened, modified or deleted,
- use of administrative rights
- directories used most frequently

Profiles are automatically generated during operation, broken down by hour. These profiles are examined for questionable behavior in each of the three categories (network, execution, and browsing). Deviations from expected behavior (over a period of one day) are computed and warnings are generated if deviations are above a threshold value.

Signature recognition is supported by the CLIPS expert system [B136]. Facts, derived from event numbers, object names etc., are used to instantiate the CLIPS signature-based rules and to identify possible illicit activity. CMDS defines UNIX attack signatures associated with, for example, failed superuser attempts, login failures, vacation activity, and attempted critical file modification. It has an equivalent set of defined signatures for the NT operating system.

The tool allows many types of report categories. Based on the above analyses, reports by user, by machine, by date, and by hour can be generated. Within these categories, various graphical displays allow visual trend analysis to be performed.

2.1.2.2 NetProwler™

NetProwler™ [C17], currently being released, comes bundled with Intruder Alert from Axent Corporation. The Intruder Alert component supports host-based intrusion detection while the NetProwler component (previously called ID-Track from Internet Tools, Inc.) supports network-based detection.

NetProwler incorporates what Axent calls the "Stateful Dynamic Signature Inspection" virtual processor. This provides a means to integrate small chunks of information being sniffed on the network into more complex events, to test events against predefined signatures in real-time, and to install new signatures while keeping the system running. NetProwler provides signatures for a wide variety of operating system and application attacks, and allows users to build customized signature profiles using a signature definition wizard. Examples of attack signatures that NetProwler supports include denial of service, unauthorized access, vulnerability probes, suspicious or malicious activity, and activity that is counter to company policies. The tool incorporates a "Profiler" that supports installation and configuration by probing the network's hosts and their applications to determine what attack signatures should be installed.

Using the attack signature definition (ASD) user interface, together with the attack wizard, users can define new attack signatures. In this way, users can characterize attacks that are composed of single events, repeated events, or series of events. An attack signature has four elements: a search primitive (a string pattern); a value primitive (a value or range of values); a reserved keyword (a protocol name such as IP); and the operating system or application associated with the attack.

NetProwler also supports automated response capabilities. This includes session logging, session termination, posting events on the event console, and alerting personnel through email, paging, and other means.

2.1.2.3 NetRanger[1]

NetRanger™ from Cisco Systems [C20] is a network-based ID system. It operates in real time and is scalable to the enterprise level. A NetRanger system is composed of Sensors and one or more Directors that are connected by a "Post-Office" communications system. Each of the Sensors is deployed on a Cisco hardware platform, but the Director is software-based.

Sensors are placed at strategic points on the network and interrogate passing network traffic. Sensors can analyze both the header and content of each packet, and can associate packets that have characteristics in common. Each Sensor can analyze single packets for attacks or can maintain state, allowing for the detection of multi-packet attacks. Sensors use a rule-based expert system to interrogate the packet information to determine the type of attack — be it simple or complex.

Three categories of attack are recognized: named attacks (i.e., attacks with a specific name); general attacks (i.e., named attacks that have spawned many varieties); and extraordinary attacks (i.e., attacks with highly complex signatures). In addition to providing many standard

1. As of 11/18/99, NetRanger is known as Cisco Secure Intrusion Detection System.

attack signatures, NetRanger provides the ability for the user to define customized signatures. In response to an attack, a Sensor has several options that include generating an alarm, logging the alarm event, killing the session, and denying further network access.

The Director provides centralized management support for the NetRanger system. In this capacity, it allows remote installation of new signatures into the Sensors, and collects and analyzes security data. The status of the Sensors can be monitored through a color coding. Entities in the system (machines, applications, alarms, etc.) have states shown as text or icons, and each state is represented by a different color. Normal states are shown in green, marginal states are shown in yellow, while critical states are shown in red. The Sensors are managed through the Director (using a Java-based tool called nrConfigure) in order to make changes to functional areas such as communications, data management, intrusion detection, and collection of sensor data.

For performance reasons, the Director does not directly support a reporting capability. This is done through third-party database support. The Director pushes out logs to a staging area from which the logging data are loaded into the database. The database can then be queried for information such as a request to plot the number of port sweeps for each day that occurred during the last week. Director notifies personnel of events via email. User-defined notifications can also be implemented.

2.1.2.4 Centrax™

Until recently, Centrax Corporation sold a product called Entrax™. However, in March 1999, Centrax was bought by Cybersafe Corporation. Cybersafe Corp. made some significant technical changes to Entrax and renamed it Centrax™ [C12].[1]

Entrax originally focused on host-based detection. Reasons included the ability of host-based systems to detect security holes in individual machines in the network, the inability of network-based ID systems to examine encrypted packets, and the limited ability of network-based ID systems to monitoring insider misuse. However, the current version of Centrax does include both host-based and network-based monitoring, a move that may have been influenced by the current popularity of the latter approach. Some reviews of Entrax/Centrax product line are included in the references in Appendix B [S24, S25].

Centrax provides two main components; a Command Console and a Target Agent. These are analogous to the Directors and Sensors in the NetRanger system. However, the Target Agent can be one of two types: one to collect host-based information, the other to collect network-

1. Paul Proctor, one of the major players in the early development of CMDS, is now the Chief Technical Officer of Centrax. These facts illustrate the great flux in the intrusion detection industry, both in terms of product names and company affiliations.

based information. These Target Agents reside on the machines which they are monitoring (e.g. individual PCs, file or print servers) and relay the information back to the Command Console for processing.

For performance, the Network Target Agent is a stand-alone machine. The host-based agents support over 170 signatures (such as for viruses and Trojan horses, object browsing, and password changing), while the network-based agents support 40 signatures.

These host-based agents can detect and react to threats locally in real time. Each threat can have its own response pattern, such as terminating the connection to the offending machine.

The Command Console performs a variety of functions through its Manager and Editing components. The Target Manager downloads auditing and collection policies to the Target Agents, the Assessment Manager probes host machines for security vulnerabilities, and the Alert Manager displays information on detected threats and can respond to these threats by, for example, logging out a user or shutting down the computer. A set of editors provides the capability to organize policies in areas such as signatures (defining under what conditions to generate alerts), audit data collection, and customized reports. The Target Agents generate raw audit data which is archived, and after appropriate reduction, sent to a database. The database can then generate reports.

The Command Console runs on Windows NT, while the host based Target Agents run on either Windows NT or Solaris-based systems. The Network Target Agent runs under Windows NT.

2.1.2.5 RealSecure

RealSecure [C15, S37] from Internet Security Systems is another real time IDS. It uses a three level architecture consisting of a network-based recognition engine, a host-based recognition engine, and an administrator's module. The network recognition engine runs on dedicated workstations to provide network intrusion detection and response. Each network recognition engine watches the packet traffic traveling over a specific network segment for attack signatures — telltale evidence that an attempted intrusion is taking place. When a network recognition engine detects unauthorized activity, it can respond by terminating the connection, sending email or pager alerts, recording the session, reconfiguring selected firewalls, or taking other user-definable actions. In addition, a network recognition engine passes an alarm to the administrator's module or a third-party management console for administrative follow-up and review.

The host-based recognition engine is a host-resident complement to the network recognition engine. It analyzes host logs to recognize attacks, determines whether the attack was successful or not, and provides other forensic information not available in a real-time

environment. Each host engine is installed on a workstation or host, and thoroughly examines that system's logs for tell-tale patterns of network misuse and breaches of security. The host engine reacts to prevent further incursions by terminating user processes and suspending user accounts. It can send alarms, log events, send traps, send e-mails, and execute user-defined actions.

All recognition engines report to and are configured by the administrative module, a management console that monitors the status of any combination of UNIX and Windows NT recognition engines. The result is comprehensive protection, easily configured and administered from a single location. The administrative module ships with either recognition engine and is also available as a plug-in module for a variety of network and systems management environments.

2.1.3 Examples of Public-domain tools

There are a number of freely available or public domain tools that can be used for intrusion detection. We reviewed two of these, Shadow and Network Flight Recorder™,[1] which are supported by a joint effort of the Naval Surface Warfare Center, Network Flight Recorder, Inc., the National Security Agency, and the SANS Institute [R39]. These systems are unlikely to have the same level of support as commercial systems, so a higher level of technical expertise is required to install and manage them. However, users are likely to better understand and appreciate how intrusion detection systems work, as well as their strengths and limitations. In addition, we review Tripwire [C22], a tool that, like Network Flight Recorder, comes in both a public domain version and a commercial version.

2.1.3.1 Shadow

Shadow [R35, B55-b] uses what it calls sensor and analysis stations. Sensors are usually located at important points in the network, such as outside a firewall, while the analysis station is located inside the firewall. Sensors extract the packet headers and save them to a file. This is read hourly by the analyst station, which then performs the filtering operation and generates a second log file. The Shadow philosophy is not to provide alerts when events are identified. This approach was motivated through experience with other ID systems, where many alerts turned out to be false alarms and were distracting and annoying.

The sensor station uses the libpcap utility developed by the Lawrence Berkeley Laboratories Network Research Group [R37] to provide a basic sniffer capability. The station does not preprocess the data, thus preventing an intruder from checking what is done with the packets.

1. The public domain version of Network Flight Recorder is no longer being supported by its developer and is no longer available at the time of this writing.

Major support for analysis is provided by tcpdump [R37] through which packet filters are defined and executed. However, some intrusions were difficult to detect with tcpdump filters, particularly those involving infrequent probes. For these types of events, Shadow provides a Perl-based tool, *one_day_pat.pl,* as part of its kit. This allows one to scan for low-frequency patterns that may occur in more than one log file. Filters can be simple or compound (Boolean) collections of simpler filters. An example of a simple filter is *tcp and dest port 23*. This simple filter selects packets with the TCP protocol and destination port 23 (i.e., telnet).

The analysis station uses a Web-based interface to display information from the sensors, or to display the results of filtering operations on the raw data. Shadow runs on many UNIX systems and on open source systems like FreeBSD or Linux.

2.1.3.2 Network Flight Recorder™

Network Flight Recorder™ (NFR) is an ID system that was previously available in both a commercial version and a public domain version. The latter was freely available until quite recently and is the focus of this section [C2, R38]. NFR explains the reasons for the change in policy in part:

> NFR has discontinued access to the old source code Research version. This was necessary as the Research software was not as capable as our commercial product and users thought the Research version was our commercial offering.

NFR plans to make the commercial product available to researchers, though apparently not in source form. We retain the discussion below because we believe that much of it is applicable to the commercial version. Readers should contact NFR for additional information.

Like Shadow, NFR uses a modified version of libpcap to promiscuously and passively extract packets off the network (this version can also extract packet bodies). However, NFR goes beyond simply analyzing headers as it reassembles TCP streams. Data collection and analysis are usually performed on the same platform outside the firewall. However, copies of NFR can also be placed at strategic internal points to detect potential insider threats. Above the packet extraction level, a decision engine filters and reassembles packets.

NFR includes a complete programming language, called N, designed for packet analysis. Filters are written in the language, which is compiled into byte-code and interpreted by the execution engine. Through programs written in N, pattern matching is performed to allow packets to be reassembled, as one example.

The functions *alert* and *record* are used to extract data after the filtering operation and to support back ends. The *alert* function can send events via email or fax. The *record* function tailors the data into formats required by the various backend analysis modules. The developers

originally thought there would be many small, special-purpose backends, but it turns out that a smaller number of multi-purpose backends were developed instead.

Histogram and *list* are two primary examples of multi-purpose backends. *Histogram* provides a facility for capturing data in a multi-dimensional matrix. Totals of relevant categories are accumulated in the cells of the matrix. Alerts can be generated based on the absolute numbers or relative frequencies within the cells. The *list* backend records chronological records, thus providing a level of detail (at the expense of storage) not provided by the histogram function.

NFR also provides *query* backends that allow you to analyze the data. *Query* backends were designed so as not to degrade the performance of the execution engine since this could lead to dropped packets. *Query* backends have their own CGI interface. Also, *query* backends provide graphical capability to allow data to be viewed more easily.

2.1.3.3 Tripwire™

Tripwire™ [C22] is a file integrity assessment tool that was originally developed at Purdue University. Like NFR, it comes in both public domain and commercial versions; the public domain version is available in source code for UNIX systems. Tripwire is different from most other ID tools in that it detects changes in the file system of the monitored system rather than looking for suspicious activities, per se.

Tripwire computes checksums or cryptographic signatures of files. If these signatures are computed for a file system that is known to be in a secure or safe configuration and stored in such a way that they cannot be corrupted, for example, offline or on a write once medium, they can be compared with a subsequent recomputation of signatures for the same file system to determine which, if any files have changed. Tripwire can be configured to report all changes in the monitored file system or only those of interest to the administrator. For example, it can check if system binaries have been modified, if syslog files have shrunk, or if security settings have unexpectedly changed. It can be configured to perform integrity checks at regularly scheduled intervals and provides systems administrators with the information they need to implement recovery if tampering has occurred. Of course, recovery requires that appropriate backups of the material that has been compromised be available.

Tripwire can facilitate recovery as well as detection — a feature not present in most ID systems. Tripwire's approach is independent of the type of exploit and it is not necessary for the exploit to be identified for recovery to be made, however, Tripwire will not detect an intrusion that does not modify monitored files and the detection will not be made until the next time Tripwire is run. Unscheduled Tripwire runs are useful for damage assessment after a suspected or confirmed intrusion.

Tripwire comes in both commercial and free versions. The current commercial release is version 2.X for a variety of UNIX platforms and Windows NT 4.0. The binary version of 2.0 for Red Hat Linux 5.1 and 5.2 is available for free. The academic source release of version 1.3 is also free and represents the state of the program as of 1992.

According to the vendor, all versions of the 2.X Commercial Release features include

- *Seamless and invisible cryptographic signing — Tripwire software for UNIX employs cryptographic signing technology. Signing the Tripwire database and policy files maintains the integrity of these critical files, preventing unauthorized changes and eliminating the need for removable media.*
- *Y2K compliance — Tripwire software for UNIX is fully Year 2000 compliant.*
- *Enhanced policy language — The Tripwire policy language has been expanded to include more flexibility in defining rules, as well as the ability to prioritize violations based upon their criticality.*
- *Email reporting — Tripwire software for UNIX sends reports via email to the appropriate systems administrator(s) based upon individual rule violations.*

Additional features are available for HP/UX and IBM/AIX platforms.

2.1.4 Government Off-the-Shelf (GOTS) Products[1]

2.1.4.1 Differences between Commercial and Government Off-the-Shelf Systems

The primary requirements for network intrusion detection products (besides intrusion detection) are to make suspicious network activity visible, provide an alert about the activity, and offer a capability to stop it if possible. On the surface, commercial and government requirements are the same; however, they generally differ in scope and focus.

In February 1999, the Department of Energy, the National Security Council, and the Office of Science and Technology Policy sponsored a workshop titled "Detection of Malicious Code, Intrusions, and Anomalous Activity." The workshop was attended by participants from commercial and government sectors. A breakout session was formed to focus on network intrusion detection. It identified the following as requirements for ID products that will NOT be met by the commercial market:

1. The content of this section was contributed by Jay Larrew of AFIWC. Because there is a lack of literature available to the public in this area, we do not have any access to reports that discuss the effectiveness of these systems and are not sure that substantive comparisons among GOTS systems or between GOTS and COTS or research systems exist. We would like to see such a comparison (Refer to Recommendation 1 in Section 5).

- *improvement in detection of attacks from well-funded, nation-state attackers*
- *intent identification*
- *objective evaluations of ID products*

Both commercial and government organizations require ID systems to detect network intrusions and provide visibility, but the scope and focus for these requirements is different.

While businesses are interested in protecting their networks and information, they are primarily interested in making a profit. Companies are motivated to purchase ID products as they become aware of the threat from network compromise, the possibility of losing critical proprietary market data or products, and the possibility of being liable for downstream damages to any trusted networks.

A second motivation is to ensure they have pursued best practices to keep from being liable to companies with which they have trusted network relationships or to stockholders in the event of loss of critical data or products. Given the bottom-line objective and the knowledge that best practices will keep them from being too liable, it is logical to surmise that the commercial market in the near term will be content with the current state of network intrusion detection technology.

Like businesses, the Federal government is interested in protecting its networks, but the primary concern is not in making a profit, but in protecting national security. Given this very important difference, governmental organizations requirement to detect intrusions has a much broader scope then the commercial requirement to detect intrusions. For the government, the capability to detect network intrusions is driven by the possibility of intruders (possibly sponsored by another country) breaking into government networks. All of the experts at the February 1999 workshop agreed that the resources and talents of a state-sponsored intruder would definitely exceed the current best practices of network intrusion detection products.

Another important distinction between the commercial and government requirements is focus. Businesses focus on gaining a picture suspicious network activity so that it can be stopped, whereas the government's focus may be to discern the intent of the person responsible for the activity. In special situations, the government may choose to collect information for intelligence purposes or for law enforcement. Businesses do not have this requirement and subsequently the vendors of best practice products available in today's market are not driven to develop this capability. Commercial ID products are designed to collect the minimum amount of data to detect suspicious activity; they are not primarily designed to collect additional data that government may need.

Government requirements alone (as articulated above) identify the need for continued work on GOTS network ID products, but there are additional arguments in favor of GOTS. The February 1999 workshop identified that the commercial market will not (on its own) develop

objective evaluations of ID products. Currently, there are no standards for evaluating network intrusion detection products, so consumers must take vendor's claims at face value. This may suffice for companies purchasing ID products to ensure they have implemented best practices (and thereby limited their liabilities) but governmental organizations required to protect national security must know both what the product does and how it works.

Another issue arises because commercial ID products can be purchased by anyone. It is possible that an intruder who knows which commercial ID product is deployed by the government could purchase the same product and learn how to defeat it. This risk is minimized by using GOTS software.

The government has network intrusion detection requirements that current COTS intrusion detection products can NOT meet. Also, the commercial market will not drive the development of the COTS intrusion detection products to meet the government requirement to protect against the nation-state-sponsored intruder, or provide the data needed by intelligence organizations or law enforcement. Given these government requirements, it is essential that the government continue to develop GOTS intrusion detection products.

2.1.4.2 Examples of GOTS Products

CIDDS

CIDDS, the Common Intrusion Detection Director System (also knows as CID Director), is a dedicated hardware/software/operating system platform supporting the Air Force Information Warfare Center's (AFIWC) Intrusion Detection Tools (IDT) program. AFIWC is the U.S. Air Force Office of Primary Responsibility for the IDT program. Within AFIWC, the Air Force Computer Emergency Response Team (AFCERT) is charged with the responsibility for day-to-day administration and network security operations involving the IDT program.

CIDDS receives near-real-time connections data and associated transcripts from Automated Security Incident Measurement (ASIM) Sensor host machines and selected other intrusion detection tools (see below). It stores this data on a local database and allows for detailed (local, regional, or theater-wide) correlation and analysis by human analysts and other automated tools.

The CID Director software consists of a suite of compiled C and C++ programs, compiled Java language programs, Oracle SQL functions, procedures, and scripts, an Oracle database structure, Borne shell scripts, and configuration files which together communicate with ASIM Sensor system machines to receive connections information from the Sensor hosts. The Director stores this information in a local Oracle database, and through the use of a user-friendly graphical user interface, allows the user to correlate, study, and analyze indicators of

potentially intrusive, malicious or unauthorized events occurring on Air Force networks. Various uses for this data include

- detecting potentially intrusive, malicious or unauthorized activities which occur over long periods of time
- detecting those activities which target specific host machines or network types
- detecting those activities that transit or involve several networks
- trend analysis and historical purposes

CIDDS also incorporates the ability to play back real-time connections data for keystroke-by-keystroke analysis.

CIDDS provides the ASIM system with a centralized data storage and analysis capability. The Director receives data from various Sensors that are monitoring and reporting on Air Force networks. These networks may be homogenous or heterogeneous, serving similar or diverse operational or support missions, and at various organizational levels. CIDDS is the central data collection, correlation and analysis point for all these network systems.

Plans for future development of the Air Force Intrusion Detection Tools system include a number of CID Directors positioned at various levels throughout the Air Force command structure, with the plan that all these Directors forward essential information to a common enterprise database in the AFCERT.

Each CID Director host machine is dependent upon the ASIM Sensor host systems reporting to it for the accuracy and timeliness of the data it receives and processes. The Director makes use of ASIM Sensor data to provide near-real-time reporting of events and provides a mechanism to review connection information in support of incident detection and event correlation. The CID Director, combined with the ASIM Sensor, provides the capability to proactively and automatically enforce Air Force network security policy and to protect Air Force networks from malicious, intrusive, or unauthorized activities.

ASIM Sensor

The ASIM Sensor software consists of a suite of compiled C code and Java language programs, Bourne shell scripts, and configuration files which together capture, filter, and analyze ethernet and Fiber Distributed Data Interface (FDDI) data packets. ASIM Sensor is essentially a promiscuous data packet sniffer and analyzer. The ASIM Sensor software monitors network Internet Protocol (IP) traffic such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP).

It then analyzes that traffic to identify suspicious activity. There are two modes of operation for ASIM Sensor: 1) batch mode, and 2) real time.

In a typical ASIM Sensor system software installation (for normal network monitoring operations), both modes are started when ASIM is installed.

Real-time ASIM uses the same software engine as batch-mode ASIM to collect network traffic. Real-time ASIM identifies strings and services that could indicate attempts at unauthorized access at the time it occurs, and immediately spawns an alert process at the Sensor host and sends a real-time alert to its associated Director. Real-time alerts typically contain basic information on a specific activity. Additional information can be viewed by generating transcripts which are keystroke-by-keystroke records of the detected alert.

Batch-mode ASIM Sensor collects network traffic (data) for a configurable time period, usually 24 hours. After collection of the data, the data is analyzed by the software to identify indicators of suspicious activity. The ASIM Sensor software generates transcripts of collected data for an identified connection that could indicate someone attempted or performed unauthorized activity. Data can be viewed at the local ASIM Sensor site or sent to a CID Director machine or central office (AFIWC/AFCERT) for review and analysis.

The data collected by ASIM Sensors at AFCERT-monitored sites is encrypted and sent to ASIM Central (AFIWC/AFCERT) each day for review by a human ASIM analyst. The analyst determines whether the activity ASIM Sensor identified is malicious, unauthorized, or normal authorized activity.

2.2 State of the ID Market

This section characterizes the current market view of ID systems. It also examines ID capabilities and looks at the ways in which organizations use ID systems. The section closes with a statement of the market condition as viewed by the authors.

2.2.1 Perspectives on What ID Systems Can Do

A great deal is being written about the capabilities of ID systems. One of the inherent difficulties for a consumer of a new, immature technology is how to interpret generalized claims of capability as they apply to the consumer's needs and how to separate hype from reality. This section includes statements from several authors in response to the question, "What can intrusion detection systems do?" These assertions are part of the market environment in which ID tools are being considered for purchase and use. They also serve to set consumer expectations.

A number of capabilities are claimed for ID products in an ICSA paper titled "An Introduction to Intrusion Detection and Assessment" [B23]. They can

- *lend a greater degree of integrity to the rest of your security infrastructure*
- *make sense of often obtuse system information sources, telling you what's really happening on your system*
- *relieve system management staff of the task of monitoring the Internet searching for the latest hacker attacks*
- *make the security mgmt of your systems by non-expert staff possible*
- *provide guidelines that assist in establishing a security policy*
- *trace user activity from the point of entry to point of exit or impact*
- *recognize activity patterns reflecting known attacks and alert appropriate staff*
- *statistical analysis for abnormal activity patterns*
- *operating-system audit trail mgmt, with recognition of user activity reflecting policy violations*

In a 1998 Computer Security Institute round table discussion of ID experts, the following perspectives were offered [B15]:

- *Realistic expectations of ID products are that they will detect common attacks in a reasonably timely manner. [Marcus Ranum]*
- *Current ID products bring the ability to view network and system activity in real-time, identify unauthorized activity and provide a near-real-time automated response. ID products also provide the ability to analyze today's activity in view of yesterday's activity to identify larger trends and problems. A good IDS will be designed to be operated at the technician level. However, it still requires considerable expertise to understand the data and know what to do in response. [Lee Sutterfield]*
- *Realistic expectations are that intrusion detection systems are discovery and detection tools that guide further investigation. An IDS deployment should have some operational procedure behind it to gather additional information and fine-tune the network and the process. A good IDS will automate as much of this process as possible. Many customers think that a given security product like an IDS will protect them from 100% of the "bad things." In a practical world, there are no absolutes, instead an IDS can significantly reduce the risk from network based threats, but they're not perfect. [Chris Klaus]*
- *You can expect to learn more about what's really happening on your network than ever before. You'll be able to gather hard data about what's being directed at your site from remote locations, and you can use that knowledge to make informed decisions about what security controls need to be deployed. [Dave Curry]*

- *Realistic expectations are that the product should detect, in near real-time, any kinds of attempts to exploit known weaknesses, or to probe your internal network. They should also keep track of attempts to overload necessary resources. Along with this, they should perhaps sound an alarm, trigger some predefined action, and keep a good log for analysis. Any existing system, or any system available in the near future, will require monitoring and maintenance by a knowledgeable and capable technical person — either as part of a remote monitoring service, as part of a local security staff, or both. Because of the uncertain nature of security policy and how to detect violations, any current or near-future system that is likely to be able to detect intrusions and misuse is also going to generate false alarms. It will require someone with enough knowledge of the environment and the nature of the IDS to sift through alarms to decide which ones are false alarms (mistakes, bugs, harmless curiosity), and which are real attacks. [Gene Spafford]*

2.2.2 Recent Survey Results

A recent (July 1999) *Information Week* survey [S30] of 2,700 executives, security professionals, and technology managers from 49 countries concludes that more companies are using intrusion-detection systems that scan the network for trespassers and alert IT personnel in real time if intruders are discovered. This year, 37% of survey respondents reported using intrusion detection products, up from 29% last year. And every company that said it uses intrusion detection systems discovered unwelcome outsiders prowling in their systems.

"That 100% of users were able to catch intrusions with [intrusion detection systems] is a testament that they actually work," says Pricewaterhouse/Coopers' Lobel. The effectiveness and growing ease of use of intrusion detection systems has helped fuel their use. "People are looking for less manually intensive and less reactive tools so they can deal with incidents in real time," Lobel adds. The tools are designed to help IT managers save time, which is important because lack of time was cited as the main barrier to implementing improved security.

	1998	1999
Alerted by colleague	47	48
Analysis of server, firewall logs	41	45
Intrusion detection systems	29	38
Data or material damage	41	37
Alerted by customer, supplier	14	15

TABLE 2-1: SOURCES OF INTRUSION ALERTS

The survey includes a numerical breakdown of responses to the question "How have you learned about your security breaches?"

The results are shown in Table 2-2. The table, in which multiple responses were allowed, implies that reliance on ID systems is growing proportionately more than in any of the other categories.

Market Sector	Percent responding
Aerospace	58
Banking/Financial	39
Communications/Telecomm	54
Consulting	42
Education	30
Government	42
High Tech/Computer	48
Insurance/Real Estate/Legal	44
Manufacturing/Distribution	42
Medical/Bio Tech	27
Military	53
Other	21
Total	41

TABLE 2-2: PERCENTAGE OF 745 ORGANIZATIONS CURRENTLY USING ID TECHNOLOGIES

The current IDS market position is further supported by the 1999 CSI/FBI Computer Crime and Security Survey [B94] of 521 security practitioners in US corporations, government agencies, financial institutions, and universities. It states that "the use of intrusion detection systems (IDS) rose from 35% in 1998 to 42% in 1999."

The ICSA/SAIC 1999 security survey (745 respondents) [S33] provides wide-ranging information about industry's attitude to computer security. Responses to one of the questions provides insights into what market sectors are implementing ID systems. Table 2-3 summarizes this data.

With respect to what market sectors will be purchasing ID systems within the next year, the following were the responses [S33]:

Market Sector	Percent Responding
Aerospace	25
Banking/Financial	42
Communications/Telecomm	32
Consulting	19
Education	25
Government	17
High Tech/Computer	29
Insurance/Real Estate/Legal	34
Manufacturing/Distribution	28
Medical/Bio Tech	23
Military	41
Other	40
Total	29

TABLE 2-3: PERCENTAGE OF 745 ORGANIZATIONS PLANNING TO PURCHASE ID TECHNOLOGIES

From these various surveys, one could conclude that ID technologies are becoming an accepted part of many organizations' information security tool suite. The concern of the authors is that such organizations are viewing these tools as solving a class of problems. The problems are not fully understood and the solutions are likely inadequate and possibly incorrect. This can create a false sense of confidence in the degree to which intrusions against an organization's critical assets are detected.

Through our own experience and in discussion with technology experts and market analysts, we observe that the current market condition of commercial ID tools and technologies exhibits a growing bandwagon effect. Each organization is referencing others in their peer group or market segment and comparing what they are doing with what the competition is doing. If an organization views itself as taking security protection actions (such as deploying an IDS) that are equal to or slightly better than an organization that it considers its peer, that is good enough. At the decision-making level, there appears to be little to no regard for what ID systems can actually do nor any appreciation for what they cannot do and should not be relied upon to do. Management's priority appears to be to ensure that they can demonstrate that they have exercised a standard of due care in the event of any legal action. We believe that the vendor community is marketing to this condition through the product claims they make. It is our observation that many of the claims made in Section 2.2.1 and the statements of survey

responders in Section 2.2.2 cannot be demonstrated in actual IDS use or, at the very least, are somewhat overstated and misleading.

2.3 What Did We Learn?

This section describes the CERT/CC® IDS team's experiences in installing, configuring, and operating several network-based ID tools on the operational CERT network, both inside and outside of the external firewall protection mechanisms. Our intent was to do what ID tool developers advised in their documentation and learn, through experimentation and use, how well each tool did what it advertised. From the knowledge we gained, we formed additional judgments to support our determination of the state of current practice. There was no intent to evaluate a specific tool or to compare one tool with another, nor was there any intent to perform formal testing of the frequency of false alarms.

2.3.1 Environment

We installed each ID tool in the CERT DMZ, which is located between the CERT private network and the CERT Internet connection. By choosing this location, we were able to monitor all traffic between the CERT private network and the Internet, between the DMZ and the Internet, between the private network and the DMZ, and between machines on the DMZ. Every machine in the DMZ is plugged into a single switch. We set one port of the switch to mirror the traffic of all of the other ports. We plugged each IDS into this port via a hub and a cable engineered to allow traffic to flow one-way only (from the switch to the IDS) as shown in Figure 2-1.

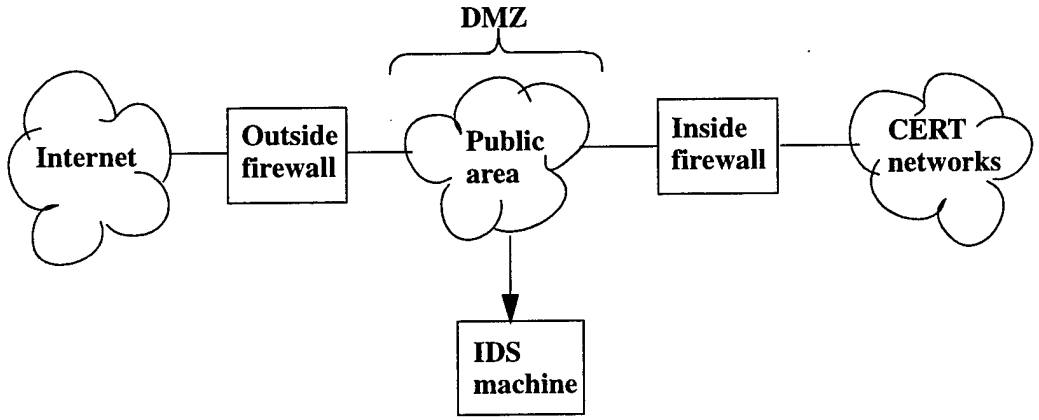


FIGURE 2-1: IDS EVALUATION SETUP

There are two problems with this approach: it isn't scalable, and it is possible to lose packets since the output from the monitored switch ports may be greater than can be handled by the port to which the traffic is mirrored (where the IDS resides). To address these problems, we could have monitored more selectively, i.e., the ports to which the inside and outside firewalls are connected. In doing so, we would have only lost the ability to monitor traffic between machines in the DMZ. However, by monitoring the switch (and all ports), we were able to tell if packet loss was a problem and were more likely to detect more events of interest. We do not recommend this configuration; however, it does demonstrate some of the trade-offs to determine an appropriate location for an IDS and what traffic to monitor.

We installed ISS RealSecure™ (Section 2.1.2.5), Cisco NetRanger™ (Section 2.1.2.3), Network Flight Recorder™ (Section 2.1.3.2), and Shadow (Section 2.1.3.1).

The policies governing our operational environment and the customized nature of the CERT infrastructure did not permit experimenting with host-based ID systems on the UNIX hosts that serve as our primary public server systems. We have tailored log filtering and analysis tools to meet our intrusion detection needs and host-based ID systems do not appear to add sufficient value for the cost and effort (primarily staff resources) required to deploy them in this environment.

2.3.2 Observations

This section describes our observations during ID tool installation and configuration.

Installation

The most important decision in installing an IDS is identifying its location(s). This decision has security implications. Most of the ID systems we examined require two interfaces: one insecure interface to perform monitoring and one secure interface to communicate with and manage the IDS.[1] We do not believe this is a reasonable approach in a production network, specifically the use of an insecure interface for monitoring, when the IDS can both read from and write to the network because there is a potential for the platform hosting the IDS to serve as a bypass for the firewall if it is compromised. If an IDS is operated with an unsecured interface, it is necessary to trust the IDS as much as, for example, the firewall and pay the same level of attention to it as the firewall requires. The approach was to secure the IDS hardware and software as much as possible and use a secure protocol such as ssh to communicate with and manage the IDS. In our judgment, the best approach is to use two interfaces as first described but ensure, through hardware connectivity and control, that the monitoring interface can read traffic traversing the network but can not write to it.

1. One tool used an alternate approach of having one interface perform all functions (monitoring, communication, and management).

The drawback to this approach is that the IDS's capabilities for automated responses that require access to the network such as killing connections and reconfiguring firewalls cannot be deployed. We do not see this as a problem for us as we would not invoke any of the provided automated response capabilities based on our lack of confidence in the ID system's ability to distinguish between actual intrusions and false alarms.

We found that commercial ID tools, in general, required less time to install than public domain tools. It appears that vendors have spent considerable time making the installation process as painless as possible. Organizations with large infrastructures should carefully consider 1) the time required to install and maintain a public domain ID solution, and 2) if the public domain solution will scale.

Configuration

None of the tools we examined had an understandable, easy-to-use configuration interface. However, the commercial tools did employ graphical interfaces where the public domain tools did not. Our primary objection is that all the tools require the operator to tune signatures individually. If the signatures were grouped into similar categories and the operator was able to tune a group at a time, the configuration process would be simplified. As an example, RealSecure documentation provides an assurance level for each signature (no false positives, few false positives, high false positives). It would have helped if we had been able to turn off ID checking for all signatures except those possessing the "no false positives" assurance level. The only way to accomplish this was to configure each signature to be "on" or "off" based on its assurance level.

Among the tools we examined, we found no indication of any integration between vulnerability scanners and configuration interfaces. The configuration process could be made more simple if detected vulnerabilities are used as the basis for the inclusion of IDS signatures, i.e., the IDS only checks for those vulnerabilities that are present and relevant to the installed system and application software (as contrasted with including signatures for software that is not present). Most, if not all, commercial vendors that sell ID tools also sell vulnerability scanners, so this level of integration could be accomplished.

2.3.3 Benefits

The majority of ID systems we examined appeared to provide good capabilities for enhanced network monitoring. However, most of the systems (both commercial and public domain) are not advertised from this perspective. Tools that are so advertised often don't provide the flexibility to examine specific packets as ID systems do.

Specifically, ID systems can monitor the following:

- firewall policy by
  - checking for IP protocol/state violations that should be caught by the firewall
  - checking for packets that the firewall is to block (i.e., confirming that "deny" rules are operating as expected)
  - categorizing, in some fashion, the types of packets that the firewall is not blocking
- unpatched machines for specific vulnerabilities. This only works if signatures are 100 percent accurate. If there is a high false alarm rate, operators will stop checking. With an open signature specification language (such as that provided with NFR), an operator can judge what the false alarm rate will be and modify the signature to change the false alarm rate, if desired. This form of monitoring is useful for
  - hosts that cannot be patched (due to, for example, the lack of a patch or one that causes other operational problems)
  - finding hosts that should be patched but aren't. In most cases, signatures are used for detecting attacks, but they can also be used to detect the version of software being used. For example, RealSecure has a signature to determine what browser is being used on different hosts and will send an alert if a host is using an old browser with vulnerabilities.
- specific network services. As networks become faster and traffic loads higher, it becomes impossible for an IDS to perform a detailed analysis of every packet traversing a network segment, however, it is possible for an IDS to monitor every packet (or at least its headers) routed to a specific port on a specific server (e.g., all DNS packets to the DNS server). Deploying an IDS in this focused fashion could be useful.

During the course of experimenting with the ID systems, we discovered one event that couldn't have been resolved without using one of these systems. Late one evening, a service simultaneously died on a number of machines in our DMZ. We looked through system logs and records but were unable to find the cause. We were able to locate the time that the services died and examined the tcpdump logs generated by the Shadow system [B55-b]. We identified the packets in the logs that appeared to have killed the services. We found the remote machines that sent the packets, and were able to get them to send the bad packets again to verify the results. With additional analysis, we discovered that the remote machines were running a version of software that produced output our local services couldn't handle. We were running the latest software version of our local services. Since the bad packets killed our services without doing any permanent damage, we were able to temporarily solve the problem by writing a script to monitor our local services and restart them when they died. We reported our finding to the software vendor and they fixed this problem. We may not have found the problem if the IDS wasn't running, and we definitely wouldn't have found it as quickly.

2.3.4 Shortcomings

We experienced two major shortcomings related to detecting intrusions using a signature-based analysis approach:

1. signature-based intrusion detection as an analysis approach
2. vendor implementations of the approach

Each category is explained below.

Underlying Issues with Signature-Based ID

Basing an ID analysis approach on the detection of known behavior patterns that exploit known vulnerabilities results in addressing symptoms, not root causes. A better use of an administrator's time and resources is to minimize or eliminate the vulnerability through the application of patches or other security measures. This focus on symptoms rather than root causes is further exacerbated by the use of signatures that detect the exploitation of known vulnerabilities in software that may not even reside on the network or host where the IDS is deployed (e.g., detecting UNIX-based patterns on a solely NT-based network). Integrating vulnerability scanners and ID systems could offer great benefits; the scanner's output could serve as input to the IDS and aid in determining what to detect (signatures selected on the basis of scanned vulnerabilities) and what to ignore (signatures eliminated due to the absence of vulnerabilities).[1]

Vendor Implementation Issues

The vendor products that we installed did not provide sufficient supporting data (such as raw packets) to test for events they claimed to detect. They also did not provide the signature algorithm that reveals how the determination of an event is made. In our judgment, this lack of information causes system administrators to spend excessive time evaluating reported false alarms, with the result that real attacks will ultimately be ignored. Given these limitations, every event requires extensive manual investigation to see whether it is real or a false alarm. For example, when a Web server exploit alarm is reported on the IDS console, the operator must look through the Web server logs to see if something actually happened. If the IDS provided all of the packets (including the responses) as supporting information, it would be fairly straightforward to see what actually occurred. Similarly, if the signature algorithm were available, the operator could determine if the signature includes checking the responses from the Web server.

---

1. In December 1999 we learned that one vendor, ISS, is working on integrating their IDS and system scanning programs.

2.3.5 Conclusions

IDS products based on current signature-based analysis approaches do not provide a complete intrusion detection solution but do produce useful results in specific situations and configurations.

We would like to see ID products provide test data to better assess the accuracy of events that are detected. We strongly support the growing community interest in open-source signatures. At this point in time, Network Flight Recorder (NFR) is the only commercial product that we would consider implementing given that the source code (n-code) for the signatures is readily available. NFR offers fewer signatures than other commercial products; however, this is being remedied in a recent agreement between the developers of NFR and L0pht™ [R88].

3 What Are the Significant Gaps and Promising Future Directions?

It remains to be seen whether or not intrusion detection technology can live up to the promise of accurately identifying attacks. Many claims are made, but few have been proven in practice. The current generation of commercial ID systems uses a limited set of techniques to detect attacks. Attackers are rapidly improving their abilities to perform successful penetration, including developing ways to defeat ID systems themselves. We see a significant number of gaps between the current state of commercial ID systems and a future state in which they provide effective defense against attackers. This section explores some of these gaps and suggests some future directions for both research and development.

First we look at some external drivers that show why alternate approaches to intrusion detection need to be pursued. Then we address how greater network complexity increases the difficulty of detecting intrusions. For example, the insecurity inherent in current network designs and topologies makes networks more vulnerable to attack. Human and organizational factors — two topics given insufficient attention to date — are addressed next. For example, the current emphasis on automated diagnosis is at the expense of any human-computer collaboration. Another factor is that of product selection in a highly volatile and uncertain market. We then identify a number of areas where the underlying technology needs to be improved. Examples include reducing the frequency of false alarms and early sensing of attacks. There are several issues associated with data analysis that need to be addressed, notably audit data reduction and the collection of forensic data to support law enforcement. Finally, we provide an overview of some of the research that is being performed to address current weaknesses. Many of these issues are identified in other documents such as "Intrusion Detection and Response" by personnel from Lawrence Livermore and Sandia National Laboratories [S14].

Although the issues in this section show that intrusion detection systems need many improvements, current ID technology, used appropriately, can satisfy many of an organization's security requirements. Section 4.4 provides guidelines to assure the most effective use of the technology.

3.1 The Need for Alternative Approaches

3.1.1 Sophistication of Attack Strategies

Gap: Attackers continue to improve their ability to penetrate networked systems and ID systems cannot keep up.

Attackers are becoming more sophisticated in their use of automated tools, in their ability to remain undetected, and in their application of unexpected strategies. Also, the frequency of these attacks is increasing. In "Cautionary Tales: Stealth Coordinated Attack HOWTO" [B80], Dragos Ruiu writes "...two years ago, we got mapped and port-scanned for vulnerabilities once a month. One year ago the scan frequency was up to once a week, and these days we get scanned several times a day with real attack attempts at least once a week." The article also states that "...the technical level of the attacks is increasing at an alarming pace, and I haven't seen any documentation of these new attack techniques." Similar observations have been made in the article "Techniques Adopted by 'System Crackers' when Attempting To Break into Corporate or Sensitive Private Networks," written by consultants with Network Security Solutions, Ltd. [B106].

The paper cited above by Dragos Ruiu [B80] identifies seven different phases of an attack, provides insights into the strategies used in each phase, and describes personal experiences defending against these attacks. The phases Ruiu identifies are reconnaissance, vulnerability identification, penetration, control, embedding, data extraction and attack relay, summarized below.

Reconnaissance: Reconnaissance entails the attacker probing the region around which a protected network operates, such as an ISP or a web site. In this phase, the attacker's purpose may be to determine the addresses of trusted hosts, to disable a trusted but less secure host, or to attack the ISP's router system. Such peripheral attacks can be given a low profile by sending the extracted information to several attacker sites and doing so at a low rate. It may be difficult for a network IDS to identify these activities since they occur outside the network's domain.

Vulnerability identification: Vulnerability identification involves discovering weak links through which the network can be accessed [B108-1]. With the automated tools available today (e.g., nmap [B129]), unauthorized scanning for vulnerabilities is very common. However, given the right detection software, the system administrator can often spot these. Web, mail, and DNS servers are examples of targets that are often exploited since there is considerable pressure to minimize down-time on the popular services provided by these targets and since these targets tend to be more exposed as they sit in front of firewalls. Vulnerability in the mail server may be particularly worrisome since the information transmitted may be confidential.

Penetration: Penetration implies defeating any security boundary perimeter such as a firewall. It can be accomplished in a number of ways. For example, the ability to execute scripts inside application programs can allow a system to be hijacked by delivering malicious code inside innocuous packages (e.g., through email). Such code can open tunnels that allow penetration through the firewall. Alternatively, a "mole" program might be installed surreptitiously by an insider that allows the same tunnel functionality. These attacks can be made to look like traffic associated with HTTP or FTP, particularly if the content is encrypted.

Control: Once an attacker has access to the network, the focus is on gaining control and removing signs of entry [B108-3]. These steps can consist of installing a small script that contacts an already compromised machine from which a larger program (e.g. Back Orifice) is downloaded. After performing these operations, a cleanup of log files (e.g., system, event files, file integrity checker files, and ID systems files) is performed by rewinding these files to their pre-attack positions. As Ruiu says [B80], "you can completely remotely control a machine and run programs on it, upload and download data, without any indication to the user other than occasional sporadic slowness — which on Windows is almost indistinguishable from normal performance, and Linux and NT aren't much better."

Embedding: In this phase, the attacker ensures that he can still retain control, even in the event that he is discovered. Since gaining control as described above puts the attacker in an exposed position, the aim here is to assure that access from this point forward does not generate obvious symptoms. This can be accomplished, for example, by overwriting little-used system files with malicious code that can survive a system reboot. Ruiu describes an attacker who inserted a piece of code that erased itself if not contacted within a specified period of time. Another insidious attack involved inserting code into an EEPROM of a network card — thus even when the operating system was reinstalled the exploit was not removed.

Data extraction: Surreptitious retrieval of information suggests that data be encrypted and trickled out at a low rate. An additional strategy is to hide outgoing data using a common protocol such as HTTP for cover, perhaps disguising the data as a JPEG or GIF file.

Attack relay: The final goal of an intruder may be to use the compromised network as a springboard for attacking other systems. Since leaving a history of attack activity from one site makes the intruder vulnerable, it is in his interest to have multiple attack relay sites. These provide the opportunity to distribute an attack, thus reducing the likelihood of being detected.

As can be seen, the spectrum of techniques available to an experienced intruder is formidable, and these are severely challenging the capabilities of today's ID systems. Improvements in detection capability are likely to be countered by new, equally creative threats.

To stay ahead of these, comprehensive measures, including more sophisticated and adaptive technology, fusion of multiple data sources, integrated human-computer diagnosis, and effective security policies and training, are needed. These and other issues are discussed in the following paragraphs.

3.1.2 Sophistication of Attack Tools

Gap: Tools to support attackers continue to improve, and are an increasing challenge to intrusion detection technology.

The intrusion detection business is growing rapidly. A major driver for this is the release of software (including operating systems) that have not been adequately tested for security. In addition, rapid increases in connectivity and data sharing leave operating systems more open to exploitation [B65]. Tools currently used by attackers offer significantly more power to less skilled individuals, greatly enlarging the user base for more sophisticated attacks. In addition, automated tools allow an attacker to penetrate and retreat quickly or to pursue slow scans which fall below most IDS detection thresholds. These factors make detection much more challenging. Scanning and remote management tools exemplify the problem.

Scanning tools allow the attacker to determine system characteristics. These tools can recognize the version of the operating system that the host is using. This enables an attacker to exploit vulnerabilities specific to the configuration. The most widely known of these tools is SATAN [B66].

Recent tools such as sscan [B99] and nmap [B119, B59] are widely used at the present time. The latest version of nmap allows identification of over 100 operating system release versions and can camouflage its attack by sending out decoy packets that appear to come from multiple innocent sites so as to hide the intrusion source. Nmap makes it much easier to survey a network slowly, at a rate that does not alert security systems [B61, B569-b2].

Port scanning allows an attacker to identify what services a host supports or does not support. Port scanning usually exploits the TCP, UDP or ICMP protocols to elicit responses from a probed host in order to identify active ports. Sscan also identifies a system's vulnerabilities. It allows one to write customized scripts that can automatically be executed to exploit the vulnerabilities that it has identified. Scanning tools such as nmap and sscan can be used just as easily to assess the security of one's own system as they can to attack another systems.

Remote management tools allow a systems administrator to remotely access and interact with servers. Significant damage can be done if such a tool is subverted. Back Orifice [B64] is a good example of a remote management tool that is commonly used as an attack vehicle. It can be installed in unsuspecting Window 95/98 computers and used to control the system from a remote client version of the tool. Originally introduced in 1998, it has been revised as Back

Orifice 2000 [B64-b2] and is now being promoted as a legitimate (and free) remote management tool. However, given its notoriety, and the fact that the distribution disk was infected by the CIH virus [B86], its utility is questionable. A competitor to Back Orifice 2000 is Netbus. It too is now released as a legitimate remote management tool, but suffers from the same legacy problems as Back Orifice. These tools can be detected by virus detectors. When in use, Back Orifice and Netbus produce network activity that should allow intrusion detection tools to sense their presence.

It is generally claimed that network ID systems have an advantage in that they are passive and therefore undetectable by intruders. However, L0pht Heavy Industries is promoting a tool called AntiSniff [C23] which "came about due to the need to determine which machines on a local network are in promiscuous mode (i.e., collecting and examining data that is not destined to them)." The stated motive for developing this tool is the detection of illegally installed sniffing software. However, it could probably detect the operation of a network IDS. It is possible that this technology could be used by an intruder to sense, and consequently avoid, a network IDS running in promiscuous mode.

New vulnerabilities will inevitably emerge, and intruders will build tools that automate their exploitation. Tools used by attackers can provide capabilities that are difficult to mimic manually (e.g., decoy and rapid-action attacks), but these attacks should be detectable using ID systems. While intrusion detection has parallels in the virus domain, intrusion detection is more complex and the field is immature. Even so, the competitiveness (and possible survival) of an IDS vendor may very well depend on the rapidity with which that vendor can provide updates that effectively defend against the latest attacks.

3.1.3 The Increasing Frequency and Changing Nature of Attacks

Gap: There is an increasing challenge resulting from the rapidly growing numbers and changing goals of intruders.

Computer intrusions have been occurring since at least the 1960s. However, with industry and government becoming increasingly dependent on networks for doing business, computer intrusions with the goals of obtaining economic/competitive advantage, political/military intelligence, and financial gain have become more prevalent. Andy Briney and Barbara Rose state in the paper "Study Confirms Increased Security Risks of E-Commerce" that "... companies conducting business online are 57 percent more likely to experience a proprietary information leak and 24 percent more likely to experience a hacking-related breach. Overall, the number of companies hit by an unauthorized access (hacking/cracking) breach increased nearly 92 percent from 1997 to 1998" [S33-5].

Figure 3.1, taken from the *InformationWeek* Global Security Survey [S30-2], summarizes the responses of 2700 security professionals to the question "Who do you suspect as the source of breaches or espionage in the past year?" There are several interesting items. First, the percent of respondents who suspect the group "computer hackers and terrorists" of committing security breaches or espionage has dramatically increased (up from about 14 percent in 1998 to 48 percent in 1999). This group has replaced "authorized users and employees," the dominant concern in 1998. This information is consistent with a conclusion reached by Harry DeMaio, president of Deloitte & Touche Security Services [S33-1]: "The internal threat has been the highest threat for so many years that it's almost a knee-jerk reaction at this stage, but as more and more remote users gain increased access to the system, the origin of the threat is changing." With the increased use of outsourcing, the rise in intrusions from contract service providers has dramatically increased (from 9 percent to 31 percent of respondents). The figure shows a significant concern in 1999 about the activities of "foreign governments," although no 1998 data was available. Interestingly, the threat from competitors is not perceived as high (about 3 percent of respondents), customers being rated a greater concern (16 percent of respondents).

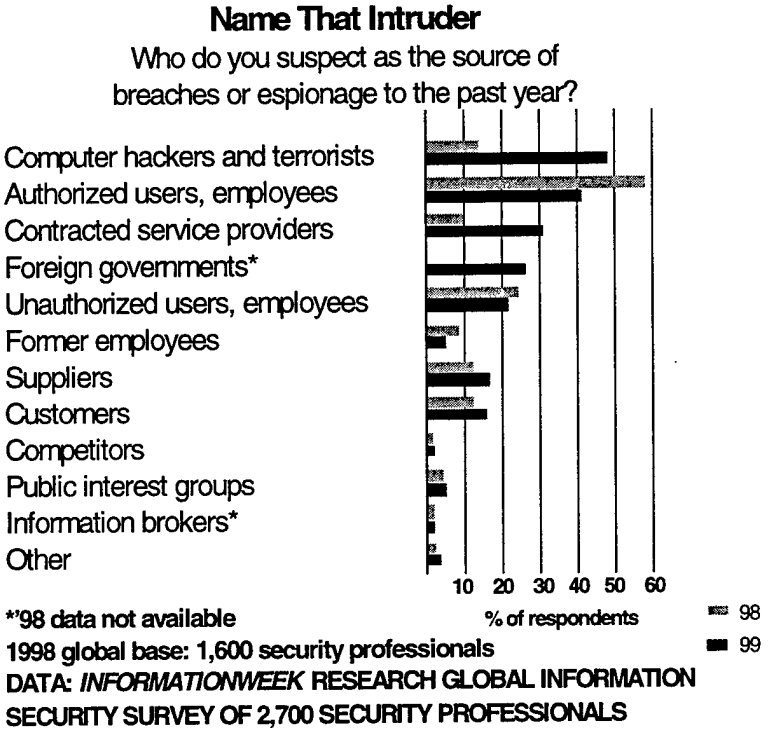


FIGURE 3-1: SECURITY PROFESSIONALS VIEWS ON INTRUDER THREAT ORIGINS, ADAPTED FROM AN *INFORMATIONWEEK* SURVEY [S30-2] (NOTE THAT MULTIPLE RESPONSES ARE ALLOWED, SO THAT TOTALS EXCEED 100 PERCENT.)

As stated in the *InformationWeek* Survey [S30-2], "Overall, security problems are becoming more serious. A year ago, half of the companies surveyed said they suffered no system downtime as a result of security breaches. This year, only 36% could make that claim." As seen in Figure 3.2, viruses appear to be by far the largest (and growing) concern, while unauthorized network entry, information loss, data integrity loss, and denial of service all compete for second place (between 10 and 13 percent of respondents). Ninety-five percent of responding organizations indicated that they use virus detection software, while about 78 percent indicated that they had installed a firewall. A surprisingly high number of respondents (48 percent) indicated they had installed an intrusion detection system. However, since data was not available in 1998, the growth rate cannot be accurately estimated.

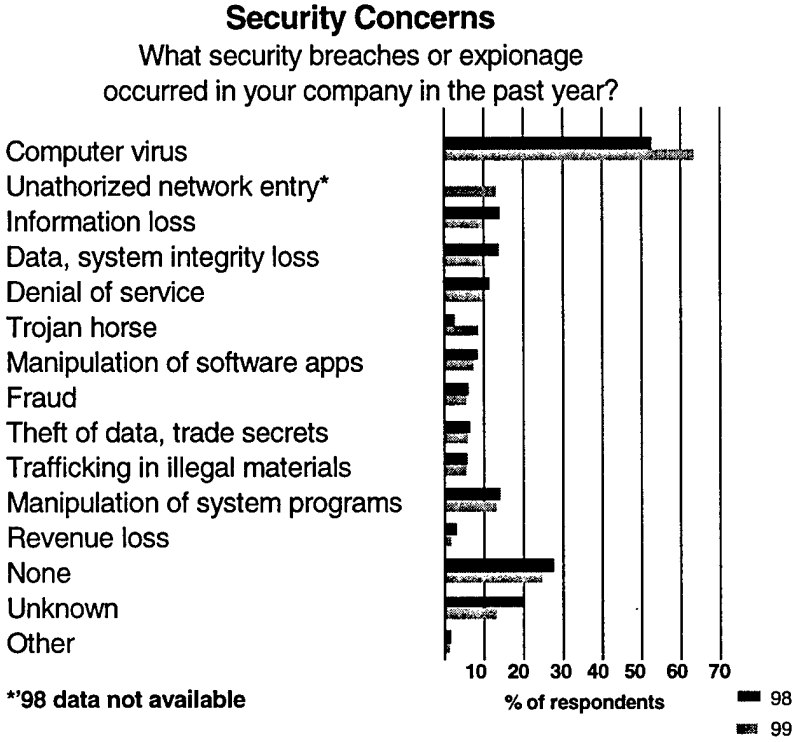


FIGURE 3-2: RESPONSES FROM SECURITY PROFESSIONALS ON SECURITY CONCERNS, ADAPTED FROM AN *INFORMATIONWEEK* SURVEY [S30-1] (NOTE THAT MULTIPLE RESPONSES ARE ALLOWED, SO THAT TOTALS EXCEED 100 PERCENT.)

These figures reveal the concerns and organizational drivers in computer and network security. Overall there appears to be a greater awareness for the need for security and a shifting belief as to who the intruders are (e.g., external attackers relative to insiders). There is also a strong concern about viruses (64 percent reported virus infections), and virus protection has become ubiquitous. The rush to electronic commerce (and its financial implications) is opening up

pathways for exploitation that were previously nonexistent. This issue alone may serve as a significant driver in motivating e-companies to install ID systems.

3.1.4 Message Encryption

Gap: Encrypted messages can contain malicious information that cannot be detected by ID systems.

Message encryption is a problem, especially for network-based intrusion systems. Encryption makes the practice of looking for particular patterns in packet bodies futile. Useful analysis can be performed only after the message has been decrypted on the target host, and this often occurs within a specific application. Driven by commercial and defense needs, message encryption is likely to substantially increase in the near future. For this reason alone, greater emphasis needs to be placed on host-based or application-based ID systems that have the ability to view message content even if the message is encrypted in transit.

While encryption may make intrusion detection more challenging, this is likely to be offset by its positive benefits. With effective encryption, theft of information becomes much harder, and the motivation to penetrate computer networks is significantly diminished. IPSEC provides a mechanism (encapsulating security payload) that can be used to hide both the contents and addresses of network packets between cooperating agents such as firewalls [B96]. However, this renders the actual source, destination, and contents of the packet opaque while they are in transit between agents. If an IDS is positioned along the agent-to-agent path, it will be unable to determine the real origin or destination of the traffic. IPSEC implementations for IPv4 exist and its functionality is part of the IPv6 [B96] standard.

Like commerce using credit cards, organizations conducting commerce through the Internet will probably come to expect and accept a certain level of abuse as the cost of doing business. The same is likely to be the case with the use (or lack thereof) of encryption. One advantage of public-key encryption is that it may reduce the amount of information that ID systems have to collect and analyze by allowing them to ignore authenticated messages.

3.1.5 Attack Strategies Targeting ID Systems

Gap: Certain types of attacks defeat the ability of ID systems to detect intrusions.

Intrusion detection systems will come under increasing threat from attacks since they represent the front-line of defense against attack. The paper "Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection" by Ptacek and Newsham [B26-b] was the first widely distributed paper to examine IDS vulnerabilities. The paper states that "...first, there is insufficient information available in packets read off the wire to correctly reconstruct what is occurring inside complex protocol transactions and next, that ID systems are inherently

vulnerable to denial of service attacks." With respect to the first problem the issue is "...that a passive network monitor cannot accurately predict whether a given machine on the network is even going to see a packet, let alone process it in the expected manner." With respect to the second point, a denial of service attack results in the IDS failing open. This implies that the IDS ceases to operate but does not prevent the rest of the system from operating or communicating with the outside world. Thus, an attacker who can shut down an IDS can then operate with relative impunity within the rest of the network.

Insertion attacks result from the IDS accepting packets that the target host rejects. In this way when the IDS reconstitutes a message from component packets, the message contains the information in the spurious packets and these make the message appear benign. However, if the target host rejects the spurious packets, the malicious string is reconstructed. Conversely, in evasion attacks, the IDS rejects packets that the target host accepts. Again, the messages that the IDS and target host see are different, and thus the attack message may get through. Unless the IDS can be set up to see exactly what is on the IP stack of the target host (a non-trivial problem), this problem will persist, and argues for the need for diverse and possibly redundant means to identify intrusive patterns. This issue is discussed further later in the context of data fusion (Section 3.6.4).

Ptacek and Newsham point out that there are two primary areas in which IDS behavior is likely to differ from host behavior. The first has to do with the handling of errors. ID systems tend to ignore checksum errors, inconsistent flags, and other phenomena that cause a packet to be dropped by a host-based protocol stack. This allows an attack to be disguised by inserting packets that will be accepted by the IDS, but dropped by the host. While it might seem simple to ensure that ID systems faithfully mimic host behavior, this is not done. The reasons stem from a variety of causes, ranging from performance considerations to ignorance of the subtle aspects of Internet protocols on the part of people who implement ID systems. The protocols themselves are ambiguous, leaving some details open to multiple interpretations so that different host operating systems can process the same packet stream with different results and still conform to the protocol. For example, packet fragmentation and reassembly combine with retransmission to create a complex situation that is handled differently in different operating systems. Unless the IDS knows how each operating system that it is protecting deals with pathologies involving fragmentation, etc., it is possible for the IDS to see an attack where none is present or to miss an attack that will succeed on a host.

Already attackers are using the above information to develop programs that will defeat intrusion detection systems. The article "Defeating Sniffers and Intrusions Detection Systems," in *Phrack Magazine* [B35] describes a variety of approaches to implement insertion and evasion attacks, and provides code to do so. The report "Bro: A System for Detecting Network Intruders in Real-Time" by Vern Paxson [R31] describes strategies for subverting intrusion detection systems. The paper classifies attacks into overload, crash and subterfuge.

The first two of these are variants of the denial of service attack, while subterfuge is related to insertion and evasion attacks.

The issues raised by Ptacek and Newsham are fundamental to the way in which network ID systems operate and are not easy to overcome. Insertion and evasion attacks require knowledge of the host for which the message is intended, and network-based ID systems do not have this knowledge [B89]. Because of the large number of hosts on a typical network, and because of the diverse applications being run on these hosts, installing host-based protection will be a major challenge. However, the alternative is the possibility that, even with an installed network-based IDS, undetected attacks will occur. We are thus likely to see increased efforts to integrate host- and network-based systems, with a focus on minimizing the installation, operating, and maintenance overhead. With such systems there is a need to integrate data from diverse sources. This will challenge both data transfer limits and data correlation (see Section 3.6.4).

3.1.6 Vulnerability to Modem Use

Gap: Intruder access using undocumented modems increases vulnerability.

Modems attached to networked machines can be points of vulnerability that escape notice [R79]. Attackers equipped with war dialers can identify and exploit such modems as points of entry into the network and can thus bypass the network's firewall and network-based IDS. In addition, security scanners do not generally include modem detection in their test suites. Because host-based ID systems analyze activity on specific machines, they are more likely to detect the presence of dial-in intruders on such machines, but only if they are configured to do so. In organizations with policies that forbid modems attached to workstations, this may be overlooked or subverted by a user who connects one.

If direct dial modems are essential, one can monitor the telephone trunk lines coming into the organization and record which lines have modem activity on them. The system can be designed so that all connectivity to the telephone network is through a single device. This allows call accounting on the device to be used for intrusion detection. Optionally, one can combine logs recorded at all interfaces between the computer network and the telephone network, using appropriate logging devices.

A solution to this problem is to make sure that no direct dial-in modems are attached to any networked machines, and that all communications go through the controlled points of entry such as firewalls or dial-back modems. Although the removal of all direct-dial modems is conceptually easy to implement, awareness of the issue needs to be raised. Thus the issue can be viewed more as a policy implementation gap than a technical gap.

3.1.7 Vulnerability to Mobile Code

Gap: There is inadequate protection against malicious mobile code.

Increasingly, executable code is being transmitted through email attachments and the use of Java applets and ActiveX objects. Given the added functionality and convenience that these capabilities provide, it is no wonder that they are popular. While ActiveX objects are less secure that Java applets, ActiveX's lack of security is reflective of its greater functionality, and the latter may win out. David Stang from Quarterdeck's Antivirus Research Center believes it is "...people's desire for style over substance. Ease of use and cool features will win over security and ill-defined security threats to privacy any day of the week" [B114]. These temptations come with considerable risk. ActiveX has a minimal security model and relies on what it calls "Authenticode" — a mechanism for verifying the code's author. This provides little guarantee of safety since "...attackers can get certificates too" [R78]. The vulnerability of ActiveX was demonstrated by the Exploder object [B113], which, on being downloaded, "benignly" shuts down a machine. Other ActiveX objects may not be so benign, for example, an ActiveX object that transfers money from one account to another without authorization [B114]. Java, which supposedly executes in an isolated environment on a target machine, is not immune as it too may have vulnerabilities [R78, B114].

Some firewalls can prevent the importation of ActiveX and Java code, through the recognition of MIME types. However, current functionality provides no ability to discriminate between legitimate and malicious code and therefore one has to either accept all ActiveX or Java code or none. As indicated in "Why Monitoring Mobile Code is Harder than It Sounds [R78], code from JavaScript (a separate language from Java) can be made to generate Java applet tags on the fly. Firewalls can identify Java code by filtering on these tags; this mechanism allows Java to be surreptitiously introduced. Some efforts are being made to develop more "intelligent" firewalls that examine the behavior of, for example, Java applets.

The examination and monitoring of mobile code and its interaction with its surroundings appears to be a good conceptual fit with the functionalities of ID systems. These capabilities go beyond what traditional firewalls provide as firewalls focus primarily on perimeter defense. The tasks that such an IDS could perform might include

- examining the contents and potential behavior of mobile code (e.g., what types of system interactions do they perform)
- tracking the execution of mobile code to assure that certain conditions are being met
- stopping mobile code if it is performing questionable activities
- alerting system administrators to the actions of suspicious code
- logging the activities of mobile code

Given the increasing magnitude of this problem and the fact that the use of mobile code is likely to explode, we believe the link between mobile code and intrusion detection warrants further exploration.

3.2 Network Issues

3.2.1 Network Size and Complexity (Scalability)

Gap: ID systems cannot yet function across environments with diverse technologies and policies.

A network domain can contain a heterogeneous collection of thousands of machines. This is challenging from a security perspective. Compounding the challenge is the fact that multiple domains may have to coordinate with each other on security matters. This complexity will inevitably increase in the future. To address the issue, several research efforts have begun to explore the scalability of ID architectures. Systems such as EMERALD [R2-a, R2-d], Hummer [R52-a], AAFID [R54-3], and NetSTAT [R30] all have such a focus. EMERALD and NetSTAT have already been discussed in Section 2.1.1, while AAFID and Hummer are reviewed in Appendix D.

Primary issues that intrusion detection faces in large heterogeneous systems are

- the integration of intrusion-related information distributed around the network(s)
  - different formats
  - different data
- sharing of sensitive intrusion-related information between cooperating but non-trusting organizations
- inter-domain coordination
  - different and possibly inconsistent intrusion-related policies
  - different intrusion detection tools
- global security in the face of failure of local intrusion detection systems

If the above research vehicles are any guide, the predominant means of addressing such issues will be through hierarchical architectures. This allows for separation of concerns, such as different policies, different intrusion detection components, and different data repositories. However, network topologies way not always be hierarchical in form, and hence more general representations may sometimes be required.

Current commercial ID systems typically consist of a set of sensor units that communicate with a central manager/analyzer unit [C10, C13, C15-a].

While these systems provide central data processing, they do not intelligently correlate data from multiple sources (see the discussion of data fusion in Section 3.6.4), and thus lack an integrated systems overview of intrusive activity. As networks become larger and intruders become more sophisticated, this type of data correlation will become increasingly necessary.

There is hope that these problems may be addressed somewhat as a result of standards activities in the ID community that are attempting to address the issues of a common data exchange format [B42-1] and the application programmer interfaces for ID systems [R9]. (See Section 3.2.3 for details.) These efforts are to be encouraged. However, given the dynamic nature of ID systems and the highly competitive market, it remains to be seen if they will be successful.

3.2.2 Lack of Inherent Security in Operating Environments

Gap: Operating systems are not designed to operate securely.

One reason that intrusion detection systems are in demand is that operating environments are not secure. In fact, it could be argued that the demand for openness encourages lax security. Peter Loscocco, et al. [B65-a, B65-b] state that because of issues such as rapidly increasing connectivity, application sharing, and use of mobile code, operating systems must place increased emphasis on security. Operating systems must be able to more effectively support policies related to access control, authentication, and encryption. While more secure operating systems have been built, these are not in the mainstream. However, it is argued that "...the threat posed by the modern computing environment cannot be addressed without the support from secure operating systems."

To meet the goals of a secure operating system, Loscocco [B65] describes several mechanisms. First, the paper describes a mandatory security policy "...that is considered to be any security policy where the definition of the policy logic and the assignment of security attributes is controlled by a system security policy administrator." Secondly, it discusses the concept of a trusted path — that is "...a mechanism by which a user may directly interact with the trusted software, which can only be activated either by the user or the trusted software and may not be initiated by any other software." Supporting the trusted path is the concept of a protected path, i.e., "...a mechanism that guarantees a mutually authenticated channel to ensure that the critical system functions are not being spoofed." The need for the operating system to support access control and encryption capabilities is discussed.

The paper suggests a variety of "gains" from the use of a secure operating system, among them being that it protects against tampering with secured applications, permits safe execution of untrustworthy applications, and limits the scope of potential damage due to penetration of applications. Through policy, a secure operating system would more tightly constrain legitimate users from accessing unauthorized data and would prevent malicious applications

from accessing data in ways that violate the intent of their user. Given the more robust security characteristics described above, the role of an ID system would be more focused and more manageable. False alarms would likely be reduced, and the need to perform under stress would be reduced.

DARPA has identified a range of "research challenges in operating system security" [R94], one of which is the meshing of prevention and detection strategies. In this paper the author suggests that

- detection techniques could be used to augment prevention techniques such as access control to operating systems
- audit data that is collected for analysis must itself be protected from tampering, and the data collection mechanisms must be protected from deactivation by an intruder

Because these are currently lacking in today's operating systems, the paper concludes that collected audit data "...is vulnerable to tampering or spoofing, and this leads to lessened faith in intrusion detection analysis."

There is a long history of secure OS development, largely funded by the DoD. Because it failed to achieve commercial acceptance, this work is generally seen as a failure. Given the competitive pressures to deliver a product's next version, security is not given high priority in design or testing. In addition, the need for products' openness and ease of use makes it increasingly difficult to incorporate security features that don't restrict the user. However, with the rapidly growing use of e-commerce, security is becoming a more critical issue, and the future demand for operating systems that are secure at a more fundamental level and provide a secure basis for intrusion detection may very well force changes.

3.2.3 Need for Interoperability and Standards

Gap: ID tools are unable to interact, making correlation of results difficult.

Both commercial and research tools tend to be monolithic, consisting of integrated sensing, analysis, and management components. Such interfaces among components as may exist are closed and proprietary. As a consequence, existing tools cannot directly interoperate, and correlating data is difficult. This prevents all but the crudest attempts to increase the confidence in alarms by looking at data from multiple sensors, to integrate data from disparate locations in a system (such as will happen with a distributed attack), or to correlate information from different ID systems. The fact that some ID systems are able to communicate with network management systems such as HP OpenView and Tivoli via SNMP does not alleviate this problem. The issue is appropriate semantic interaction among systems rather than control of individual ID systems.

The lack of interoperability has been recognized by both the vendor and research communities each of which is addressing it in their own way. Under the prodding of DARPA, the research community is developing the Common Intrusion Detection Framework (CIDF). Within the Internet Engineering Task Force (IETF) is the Intrusion Detection Working Group (IDWG). Although vendors were invited to participate in the CIDF effort, most chose not to do so. On the other hand, the IDWG is primarily a vendor oriented group, though it appears not to be a typical IETF working group.

CIDF is developing the Common Intrusion Specification Language (CISL). This seems to be a fairly heavyweight attempt to provide a basis covering all conceivable ID architectures, but comments at a recent DARPA Principle Investigators' meeting indicate that several research groups have found a need to develop their own specification languages which may or may not be extensions of CISL. DARPA has established a working group to discuss the issue of intrusion specification languages, but it is not clear whether this group will make a significant contribution in the area as, to date, activity has been minimal. There appears to be little interest in the CIDF approach among IDS vendors, and it is unlikely that this effort will have any near term impact on the commercial world.

The IDWG is primarily a vendor group operating within the IETF framework. Currently, it has a requirements document for intrusion detection message exchange and proposals for protocols, a data model, and data representations within the framework established by the requirements document. If the normal IETF process is followed, these proposals will be accepted by the group, and trial implementations will be produced by one or more participants. Once interoperability among multiple versions is demonstrated, the proposals can proceed towards standardization as one or more IETF request for comments (RFCs). This will probably require a year or more to occur.

The IDWG approach is much more modest than that proposed by CIDF. Presently, only alert data is proposed for interchange and the amount of information required to be transmitted is minimal. Appendix E contains further information about these two efforts.

3.2.4 Inherent Limits of Network-based ID Systems

Gap: Network-based ID systems may not see all traffic.

Network intrusion detection systems analyze the traffic passing though network segments in an attempt to detect attacks on both the network infrastructure and on the attached hosts. In the former case, classes of attacks that would be difficult to see from the perspective of a single host can be seen easily at the network level. In the latter case, the network IDS must make sure that its view of what a given host will see is consistent with that of the host. There are a number of factors that make both situations problematic.

Recent years have seen a trend towards switched, as opposed to broadcast, local network segments. This trend means that there may not be any location within an enterprise from which both internal and external traffic is visible. Careful analysis of the network configuration is required to determine whether or not a network IDS is capable of seeing all the traffic of interest. This trend is further exacerbated by the growing use of virtual private networks and encrypted tunnels which render network traffic opaque to the IDS. Even SSL Web connections can provide cover for some forms of intrusions. In addition, the bandwidth of both internal and external network connections is rising rapidly. Multi-megabit connections to the outside world are becoming commonplace. One hundred megabit internal Ethernet connections are the rule with gigabit segments offered by most router and switch vendors. Network-based ID systems are less and less able to keep up with these traffic loads. Router vendors whose backplanes must pass these traffic levels are unable to allocate the computational capacity necessary to perform stateful examinations of this traffic. At traffic levels that permit analysis, many intrusions require the accumulation of substantial state information for their identification. Examples include the creation of probe or scan histories, tracking of incomplete protocol exchanges, such as TCP opens, reassembly of fragmented packets, etc. This opens the IDS to denial of service attacks based on exhaustion of its internal resources. These trends will most likely continue for the foreseeable future. As a consequence, network-based ID systems will have decreasing applicability and will have to be replaced by alternative approaches.

When a network IDS attempts to infer the effect of traffic directed towards a specific host on that host, additional problems arise. Depending on the position of the IDS in the network, it may be impossible to ensure that the traffic seen by the IDS and the hosts that it is protecting are the same. If there are multiple paths between a host and the outside, only some of which pass by the IDS, it is possible that the vagaries of Internet routing may send some packets between a given external source and the target host through routes that are not visible to the IDS. A coordinated attack from multiple hosts may make this even more likely. Even if the IDS sees all traffic that is delivered to the host, there is always a possibility that the host will drop packets due to resource limitations within its own protocol stack. It is not feasible for the IDS to model the behavior of all hosts on the network with sufficient accuracy to avoid problems of this nature.

3.3 Human and Organizational Factors

3.3.1 Cooperation between Organizations that Lack Complete Trust

Gap: Mechanisms are lacking that allow competing organizations to cooperate for their common good.

Independent organizations may need to cooperate with each other on computer intrusion issues. For example, organizations could be members of a competitive community that, out of enlightened self-interest, need to cooperate on common security issues. In some cases it may be necessary to transfer potentially untrustworthy information between such organizations or to query or notify one another about intrusion-related incidents. Alternatively, there may be a need to share vulnerability-related information within an application domain.

A recent paper, "Research Issues in Cooperative Intrusion Detection between Multiple Domains" by Frinke, et al. [R66], investigates data sharing between entities that have different security policies and incomplete trust. The work focuses on the issues of authentication (Who sent the message?), reliability (Is the data accurate?), and integrity (Is the received message what was sent?) of intrusion-related information. The tool, Hummingbird, which has been used to explore these issues, is reviewed in Appendix D. This issue of untrustworthy communication is briefly alluded to in the SRI paper "Future Directions for Intrusion Detection" [R59].

Research at Columbia University has been addressing the issue of sharing models of fraudulent transactions within the financial community [R53]. Clearly, financial data can be highly proprietary, and there is a great reluctance within that community to admit to being a victim of financial fraud. However, in order for the financial community to defend against fraud, they must cooperate and share some security-related information. The approach relies on cooperating financial organizations developing local fraud detection models (using a pattern-directed inference approach), and then combining these models across organizations in order to generalize the detection capability. This approach allows an abstraction of the financial data such that proprietary information is removed while still retaining the ability to share essential intrusion-related information. The research vehicle for this work is JAM (Java Agents for Meta-Learning; see Appendix D).

Broad coordinated attacks against computer networks of organizations that share similar characteristics and roles (e.g., financial institutions, military agencies, utilities) will become more frequent. To combat these attacks, mechanisms such as those described above are helpful in supporting coordination and cooperation among potential victims. While there is a clear advantage to such cooperation, many organizations (especially in the financial industry) are

very sensitive about making public statements on issues pertaining to their security. Recent proposals for a nation-wide intrusion detection network (FIDNET) put forth by the Government Services Administration immediately ran into a storm of criticism over such issues. The success of these efforts is therefore likely to depend more on each organization's ability to cooperate with its peers than with any technical issues.

3.3.2 Need for Greater Human-Computer Interaction

Gap: Human skills are not adequately used to support diagnosis of intrusions and to determine resulting actions.

People excel in detecting patterns, particularly if the basis for these patterns is not fully understood (e.g., people have hunches, a sixth sense, a feeling in the gut). We have strong skills in identifying unexpected relationships between entities, and use analogy to recognize new relationships. Symbolic reasoning and graphical thinking are central to these abilities (see Section 3.6.6). On the other hand, computers are able to search through, compute on, organize, and graphically display massive quantities of data without introducing the types of errors that people are prone to make. These capabilities complement the human skills.

Since the detection and diagnosis of an intrusion can be challenging, it may take diverse points of view to reach a reliable conclusion. Combining the complementary analytic strengths of a person and a computer to reach reliable conclusions thus makes sense. To date, however, this symbiosis has been marginally exploited [C4]. Commercial network-based ID systems indicate that an intrusion has been detected and provide only low level data to support conclusions. They offer no information to help determine if a conclusion is credible with respect to the evidence or diagnosis.

There is little opportunity to examine the reasoning behind the conclusion or to interact with the system in order to examine alternatives. Many security managers will not want to interact with an IDS in this manner — they would rather be handed the result — but this can only lead to a false sense of security. However, with many types of intrusions, the probability of a correct machine diagnosis may be low, and operator involvement becomes increasingly necessary if a correct outcome is desired. The issue of correct diagnosis is critical early in an attack scenario (see Section 3.4.1) when subtle information may only hint at a problem. In this case, human insight and experience may be crucial to assure a positive outcome.

Of particular interest is Rasmussen's model of operator mental activity [B71]. This model suggests that operators of complex equipment structure their cognitive reasoning into three domains related to skill-based behavior (when human response is relatively automatic), ruled-based behavior (when rules, known through the operator's past experience, are applicable), and knowledge-based behavior (when there is sufficient uncertainty that human reasoning from first principles must be performed). These three regimes directly map to the regimes in

which an IDS can automatically determine the nature of an event, when there is some standard interactions between the operator and the IDS, and when the IDS must rely significantly on the operator to diagnose the problem. The latter case directly relates to the detection of a new type of intrusion.

There is a considerable body of research on human-computer interaction [B71, B72, B73], and this discipline appears to have significant, though unrecognized, relevance to the technology of intrusion detection. We believe that this fruitful research area needs to be aggressively explored.

3.3.3 Identifying Intruders Through Profiling

Gap: ID systems are unable to identify attackers and their goals.

In criminal cases (particularly serial murders), profiling perpetrators based on evidence is an accepted technique. Analogously, the evidence left by an intruder on a computer network provides evidence, both as to who the perpetrator is and (perhaps) what his goals are. The tools and techniques an attacker uses, and the manner in which they are used (e.g., frequency of use of certain commands or command sequences), provide a fingerprint that is probably unique to an individual. While profiling legitimate users has not met with much success, it is not intended that this type of intruder profiling be automated. Rather it is a computer-supported but manual task that attempts to identify the idiosyncrasies of the attacker's behavior.

Trying to directly identify an attacker's goals could be difficult since a direct inference between actions and goals is likely to be weak. For example, an attacker's goals might include either curiosity or information theft. Curiosity may be a mild irritant. However, the consequences of information theft may be the loss of a crucial competitive advantage. Based on observing only the symptoms of the intruder's activity, it may not be possible to distinguish between the two. Nevertheless, in other cases, where the physical outcomes are different, some weak inferences could probably be made between activities and goals.

Given the ability to identify an attacker from his prior methods of operation, and given an understanding of the attacker's prior goals, it may be possible to infer where his current activity is leading. For example, an intruder who, in previous attacks, was after military secrets will probably continue in this vein, while an intruder who was vandalizing an ex-employer's database will probably continue along this line. Insights into the attacker's goals would be invaluable since they provide a basis for action. Actions might include continued passive observation, attempting to trace the attack's point of origin, blocking the intruder's progress, locking him out of the network, or providing a honeypot for entrapment [B108-4].

Attacker profiling for intrusion detection has not been explored to any great extent [R80, R81], but might have promise. Much historical information on intrusive activity has been gathered within security support organizations such as CERT/CC. This data could be mined to determine whether intruder profiling provides an accurate characterization of individual attackers. Additional information may be found in documented case histories [B19, B27, B28, B76, S19]. Note that this area of investigation may provide significant support for forensic analysis (see Section 3.5.3).

3.3.4 Taxonomic Foundations for Intrusion Detection

Gap: Widely accepted ID terminology and conceptual structures are lacking.

The nomenclature used by researchers and practitioners in the field varies widely within the community. Indeed, this variation in terminology has motivated the authors of this report to develop a set of definitions for technical terms used within the report (Appendix A). SANS has developed a set of terms for use by the ID community [B6].

There have been a variety of attempts to organize the field's conceptual structures, especially taxonomies defining types of intrusions and intrusion detection techniques. Some of these are identified in Appendix D. Of particular importance is the need for a taxonomy of vulnerabilities, as this would provide a more rigorous basis on which to develop intrusion detection methods. A simple categorization of vulnerabilities by Kragen Sitaker can be found in "How to Find Security Holes" [B100], while a paper from Matt Bishop [R82] discusses vulnerability analysis as a precursor to classifying vulnerabilities. The latter paper describes his initial efforts to construct a classification. In addition, there is a recent effort, hosted by MITRE, to sidestep the taxonomy and develop a comprehensive listing of exploitable system flaws known as CVE[1] [R83]. The rationale for this effort is given in a paper [R83-b] that appeared near the beginning of the CVE effort.

> There are several efforts that attempt to define shareable schemas, universal primitives and robust taxonomies for vulnerability information. While we remain interested in and supportive of these efforts, we note that there appears to be no consensus on these issues yet. Our problem is that we need to achieve interoperability among our tools right now. Moreover, the problem preventing us from achieving this integration is much more basic than the definition of good schemas, primitives or taxonomies. The problem is that there is no consistency in the community with regards to identifying the vulnerabilities.

1. CVE originally stood for Common Vulnerability Enumeration. In August of 1999, Steven Christey of MITRE suggested that the name be changed to Common Vulnerabilities and Exposures in response to concerns that the term "Vulnerability" might be considered ambiguous [R83-c]. Although there is no record in the archives of the CVE editorial board adopting the change, it seems to have stuck.

To be fair, this is characteristic of an immature and developing field as standards, especially for terminology and acceptable practices, are typically codifications of widely accepted practices.

Within such a young, dynamic field as intrusion detection, it may be some time before concepts and their relationships to each other become stable. However, if an effective taxonomy of vulnerabilities could be developed, it would provide a more solid basis for reasoning about intrusion from a defender's point of view, and would provide a more rigorous foundation for building intrusion detection systems.

3.3.5 Rapidity of Change in the Commercial Market Place

Gap: The volatility in the ID marketplace makes the decision about which product to purchase and maintain difficult.

The commercial intrusion detection field is rapidly changing with respect to

- technical capabilities of products
- emerging new products
- consolidation of products
- consolidation of companies
- emerging new companies

This bewildering sea of change makes it very difficult for the buyer of products to make rational purchasing decisions. During the course of our investigation, we have seen almost constant change. Because of the rapidity of this changing environment, we have had to rely on web material for a substantial portion of the information that appears in the report. In several cases, reports that appeared in one location early in the process were either moved to another location or disappeared before we finished. In at least one case, the company responsible for seminal work was acquired by another; reports were moved, then removed. Another product was spun out of its original home as a separate company which was subsequently acquired by a third company.

Several of the products that we attempted to deploy underwent revision while we were experimenting with them. Although we recommend a comparative evaluation of commercial products in the ID arena, some products will likely become obsolete during the process. Therefore, evaluation may need to be an ongoing process to be useful.

These changes are likely to continue and provide a further indication of the immaturity of the field. Because this process is largely unpredictable, it is difficult to give advice about how to deal with it. Although we see continuous change, the change does not appear to represent

significant breakthroughs in either technology or usability. The best strategy may be to make purchasing and deployment decisions based on the products and information available at decision time and not to look back or to allow second guessing of the decision. At the same time, users should realize that change is inevitable and that improvements are made, so that purchasing and deploying a new IDS periodically may be necessary.

3.4 Functional Issues

3.4.1 Sensing an Attack Before Damage Occurs

Gap: ID systems are unable to identify many types of intrusion in their early stages.

The more lead-time an IDS can provide in identifying an attacker's activities, the better the chances that damage can be averted. Thus it makes sense for an IDS to know the early signs and sequences of activity that characterize an attack. For detecting signs that characterize an attack, statistical profiling has been shown to indicate hostile probing activity. For detecting sequences, work has been performed on developing temporal models of attack sequences. In addition, implementation of manual intrusion detection policies can be effective to support early warning. These approaches are described below.

Attackers who are not familiar with a network generally probe the system to acquire information on the system's characteristics (See Section 3.1.2). It is argued in "Towards Trapping Wily Intruders in the Large" [R81] that there are significant statistical differences between the use of network services for legitimate purposes and the use of network services to support intrusion. For example, while the presence of TCP resets may or may not indicate hostile probing, the characteristics of suspicious use are easy to identify. While this technique does not detect many intrusive behaviors, it could be a useful early warning indicator.

A more general strategy is to model attack scenarios. In this approach, there are notions of attack sequences that lead to compromised configurations. As a hypothesized attack evolves, evidence is built up for and against the probability of that type of attack based on predefined attack models. This approach was first proposed by Garvey and Lunt [R23] and later implemented in the STAT methodology [R3, R4, R5], a state transition approach. It allows the IDS to provide early warning of an attack, with growing or diminishing confidence of the existence of the attack as the evidence unfolds. (See the review of NetStat in Section 2.1.1.2) A similar approach uses colored Petri nets to capture the attack sequences [R7]. (See Section 3.6.5)

While tools can be of significant benefit in helping to detect early signs of intrusion, enforcement of manual policies can greatly help this area and does not require major investments in technology. By frequent review of logs, processes, and other network/system

information early activity can likely be spotted. The SEI reports *Preparing to Detect Signs of Intrusion* [B116] and *Detecting Signs of Intrusion* [B98] provide practical advice to system administrators on what to look for in this area. (See also Section 3.4.3.)

More research needs to focus on identifying early indications and warnings of an attack. The more an intruder has penetrated a network the more difficult it is to assess the state of the system and to know how much damage has been done. With effective early warning, much of the need for complex tools to deal with eliminating the penetration and recovery from damage is removed. This is an area where human experience and insight could play an important part in guiding early, but probably unreliable, computer-based analysis.

3.4.2 Automatic Response to Intrusion

Gap: ID systems are unable to take appropriate automatic action in the event of an attack.

Automated response to attack (i.e., aimed at recovery or defense, not reprisal) is appealing since

- it does not require continuous human oversight
- it can act more rapidly than humans
- it can be tailored to, and will consistently follow, specified policies

Some current commercial ID systems do support operator intervention. For example, CMDS supports four response levels of manual intervention: ignore the warning, increase observation, denial of access, and emergency shutdown [C13]. RealSecure™ allows an associated firewall to be reconfigured to reject traffic from a specific IP address [C15-a1]. These actions do require a human operator to be in the loop.

Given the current maturity of IDS technology, the dangers of automated response are significant, and outweigh the above advantages. With the frequency of false positives that exists in the current generation of ID systems, the potential for inappropriate response to misdiagnosis is too high. In addition, automated response could be exploited by a perpetrator whose aim is to induce denial of service by spoofing an attack from a legitimate user. Finally, as indicated in the article "Intrusion Detection and Response" [S14], the issue of limiting the effect of automated response so as to prevent cascade and livelock[1] failures that may be caused by the response system needs to be addressed. While recognizing the dangers of automated response, the same article suggests that automated response is clearly necessary, especially in critical infrastructure elements involving speeds or volumes beyond the human

1. A situation in which some critical stage of a task is unable to finish because its clients perpetually create more work for it to do after they have been serviced but before it can clear its queue. Differs from deadlock in that the process is not blocked or waiting for anything, but has a virtually infinite amount of work to do and can never catch up. [B81]

capacity to respond. However, until the accuracy of ID diagnosis is significantly improved, appropriate automated response remains high risk.

There is a clear need for ID systems to be able to distinguish between scenarios that are absolutely guaranteed to indicate invalid usage and every other kind of scenario. There are a large number of these "absolutely invalid" scenarios. One additional way to define "absolutely invalid" is that the victim is willing to take programmed actions in response to their occurrence. If the victim is unwilling to take such actions, the scenario by definition is not "absolutely invalid."

3.4.3 Post Intrusion Activities — Recovery and Reprisal

Gap: ID systems provide little to no support for damage recovery.

After an intrusion has occurred and damage (e.g., data corruption, data modification, information theft, or system compromise) has resulted, can intrusion detection systems help? Current ID systems provide little direct support for damage recovery [S14, B67]. Commercial network ID systems most often rely on simple string matching to detect known patterns of intrusion. This provides neither any indication of the extent of the damage nor how the damage could be repaired. Host-based intrusion detection systems such as Centrax [C12] may be better candidates for supporting system recovery.

One tool that currently does provides meaningful support for post attack recovery is Tripwire [C22]. This file integrity-checking tool was originally developed at Purdue University and later commercialized. The Tripwire Security Systems Web site states: "Tripwire monitors all servers and clients on a network, detecting and reporting any changes to critical system or data files. Tripwire can absolutely, unequivocally determine if a protected file has been altered in a way that violates the policy set by the administrator" [C22]. Tripwire creates cryptographic signatures for all critical system data and stores the signatures in a protected database, preferably off-line or on read-only medium. Upon administrator request, Tripwire recomputes the signatures and compares them with those in the database, detecting any changes called out in the Tripwire configuration file. If undesirable changes are detected, the system can be restored to its original configuration. A more limited response of this type is provided by Network Associate's CyberCop's tool. This provides a "AutoRestore" capability that can delete Web pages corrupted by an intruder and can replace these with the original pages.

As described in Section 3.3.2, much can be accomplished through manual support of recovery activities. The SEI report *Responding to Intrusions* [B123] provides information in this area, particularly within the context of developing a recovery policy. In addition, the article "Some Tips on Network Forensics" by Ranum [B68] provides some practical insights about intrusion response.

A controversial issue is that of reprisal against perpetrators. An IDS can provide information on where a detected attack is emanating from (or at least where the attack appears to be emanating from) so that "attack-back" could be performed [B44]. However, if the intent is to damage a perpetrator's capability (as opposed to sending a warning), the dangers are many. Given the problems of, for example, spoofed IP addresses and false diagnoses, retaliation is more likely than not to hit the wrong target. This is further complicated by the fact that wireless phones can make it much more difficult to track the ultimate source of the attack. In the event that the attacker is successfully identified, the victim better be sure that his defenses are secure and that the attacker does not have a backdoor to his system. Counter-reprisals could be devastating. Approaches to mitigating this problem through an improved ability to track perpetrators are discussed by Cohen in "Providing for Responsibility in a Global Information Infrastructure" [B69]. While the IPv6 standard [B88] provides some intriguing possibilities with regard to intrusion protection through authentication, it is unlikely that this protocol will deter a serious attacker.

Recovery can be significantly simplified if one has enforced policies and procedures in place to perform frequent backups (to a write-once device or removable medium) and has mechanisms in place to minimize the likelihood that backups are not corrupted (using, for example, an integrity checker). However, with an experienced intruder, subtle modification of files or theft of information may be difficult to detect. Information theft may be especially challenging because all evidence of the intrusion can be removed if audit trail information is stored on an unprotected disk. In situations where security is critical, it therefore makes sense to store audit records on write-once media on a physically secured device. Because these records can be voluminous, compression of audit records may be necessary (see Section 3.5.2).

3.4.4 Performance or Lack Thereof

Gap 1: ID systems cannot keep up with normal traffic loads.

Gap 2: ID systems cannot detect and counteract intruders that are starving the ID systems of resources.

To be effective, a network ID system must be able to analyze all incoming packets. If the IDS cannot keep up with the throughput, then intrusive patterns may not be seen and a vulnerability is created. In evaluating ID systems, a NSTL survey states "...sure, ID systems spot attacks as advertised-on empty networks. They also work well on heavily utilized Ethernet segments. But fill up a fast Ethernet segment with traffic and that vigilance vanishes; in fact, no product detected all the attacks when the network was heavily loaded" [S20]. This comment is consistent with the concern raised in a CSI roundtable discussion [B15] which likewise indicates that "...most IDS products can't even keep up with 10Mbps Ethernet speeds. The

networked environment is rapidly moving way beyond that speed." Thus an attacker may bypass an IDS out of "dumb luck" simply because of the current high traffic.

The premeditated use of resource exhaustion to attack or avoid an IDS is discussed in some detail by Ptacek and Newsham [B26-b]. Three types of resources that can be exhausted are identified: CPU cycles, memory, and network bandwidth. In the first case, Ptacek and Newsham state, "... the IDS spends CPU cycles reading packets, determining what they are, and matching them to some location in the saved network state... An attacker can determine what the most computationally expensive network processing operations are and force the IDS to spend all its time doing useless work." Regarding memory exhaustion, Ptacek and Newsham [B26-b] state, "...an attacker can create a stream of meaningless events and, by having them continually stored, eventually exhaust all disk space on the IDS which will then be unable to store real events." Finally, with respect to network bandwidth, they further state, "...an attacker can overload the network with meaningless information and prevent the IDS from keeping up with what's actually happening on the network." The underlying problem is that a network IDS must analyze all network traffic moving through it, while most other components only have to deal with subsets of this data. Thus, the IDS is more likely to be bottlenecked. In a sense, some types of ID systems make easy targets for resource exhaustion attacks because they devote resources to maintaining state information about partially completed attacks on multiple hosts. Unless the resources of the IDS have almost the same aggregate resources of the protected network, the ID system may be quite vulnerable. Thus, if the IDS is the only means of protection available, a successful attack on the IDS may leave the rest of the system vulnerable.

High growth in future network traffic will make the matter worse. To address this issue there is a need to

- make sensors more intelligent so as to reduce the quantity of information that they send to the central ID controller
- make sensors more efficient so as to perform complete analysis without dropping packets

3.4.5 Identifying Unknown Modes of Attack

Gap: ID systems are not able to recognize new attack strategies.

ID systems that rely on matching patterns of behavior that represent signatures of known attacks are unable to detect previously unseen attacks with different signatures. ID systems need to protect against unknown or new attacks; the following approaches may be worth pursuing:

- detecting changes in system configuration and file content

- detecting actions that are outside those allowed for privileged programs
- detecting deviations from known user or group profiles
- having a notion of what constitutes normal system behavior and detecting deviations from this

The first of these is discussed in Section 3.4.3 and provides an approach that is based on existing technology. One weakness of this approach is that the perpetrator must already have compromised the network to the extent that files or system parameters have been modified. Even if a backup of the environment allows one to revert to a prior valid configuration, sensitive information may be lost and/or stolen.

Privileged programs (such as UNIX sendmail and fingerd) can be vulnerable to abuse through the execution of root-level commands that were not intentionally designed into the programs. Work by Ko, Fink, and Levitt [R96] has explored mechanisms to detect violations of the legally specified behaviors which, for the programs examined, are quite simple. Thus, any behavior which is outside the allowable set can be detected. This approach is as applicable for unknown exploits as it is for known exploits.

The third approach is the oldest — namely anomaly detection through user profiling. This approach was originally proposed by James Anderson [B34] who hypothesized that "it is possible to characterize the use of a computer system by observing the various parameters available through audit trails, and to establish from these observations, 'normal' ranges for the various values making up the characterizations." Statistical profiles of legitimate users are stored, and these are compared to the behavior of current users. If the current behavior deviates significantly from the behavior reflected in the stored profile, an anomaly is indicated. This approach was also pioneered in the early 1990's by SRI International [R1-a, R1-b, R1-c, and R1-d], and was incorporated into at least one commercial system [C11]. However, this approach has never seen widespread adoption since

- defining parameters that accurately characterize "normal" behavior is difficult
- user behavior can change too rapidly for the learning system to adapt to this new behavior, and such deviations could result in false positives
- sophisticated intruders could, in theory, train the system to accept their behavior as normal

The fourth approach to detecting unknown attacks is to determine what represents normal behavior within networks, hosts, or applications, and to flag activity that does not reflect what is expected. This is an extension of user profiling to protect other infrastructure assets. One approach was originally inspired by biological immune system principles that have a "sense of self" [R50]. It was hypothesized that many of the attributes of the immune system (e.g., recognition of foreign bodies, self-repair, and no trusted components) could be transferred to intrusion detection. In an initial investigation [R48], short sequences of system calls, resulting

from the execution of normal privileged processes, were stored in a database. These represented the "self," and were compared with system calls from process sequences that contained abnormal behavior. At least for simple cases, it was shown that such abnormalities could be detected. The notion of "self" has been extended to look at distributed CORBA™-based networks [R60]. In this example, a CORBA application is considered as the organism whose cells are the clients within that application. The focus of the research is to determine if "rogue" events can be detected after training the system to recognize normal behavior, as reflected by system call sequences. In both examples, the results were encouraging, but the authors recognized that additional work was necessary before these notions of "self" could be translated and applied for intrusion detection. Further details on "computer immunology" can be found in Section 3.6.1 and Appendix D. While encouraging, this approach is still experimental.

If a method of attack is unknown, then only indirect evidence of its presence can be detected, and this makes the intrusion detection problem hard. The techniques described above attempt to infer novel intrusions, but, with the possible exception of file integrity checkers, success has only been on an experimental scale. Use of an immunological analogy is, in principle, very powerful. However, it remains to be seen if system call sequences (i.e., the entities that provide a notion of the computer's "self") provide a sufficiently unique discriminator of authorized and intrusive behavior. Underlying any technique that attempts to protect against unknown attack is the need to provide an IDS with a notion of "self," since anything that is "not self" is by definition anomalous. Unfortunately if the notion of self is not sufficiently accurate, too many anomalies will be detected raising the number of false positives to an unmanageable level.

3.4.6 Providing Guidance on Actions Resulting from Diagnosis

Gap: ID systems provide little to no guidance for responding once an attack has been identified.

Current commercial ID systems provide little guidance about appropriate responses when attacks are detected. Because of the high frequency of false alarms, vendors may be reluctant to give advice. Multiple diagnoses can result from a set of symptoms, and it would be confusing if different guidance was given for each of these. However, there may be intrusions where the symptoms are unambiguous, and where the consequences of not acting are serious. In these cases, some guidance could be extremely valuable.

In the competitive world in which ID systems exist, there is a reluctance to divulge proprietary signatures that represent known patterns of attack. This includes the publication of information that reveals inaccuracies of signatures. With a library of openly available signatures, the problem would be reduced. The ID community would be free to scrutinize the logic of the

signatures and to provide improvements in their precision (i.e., the consistency with which signatures would predictably identify known attack patterns). Precision would help ID users determine the best response and give them a high degree of confidence that it was appropriate.

3.4.7 Signatures and Their Accuracy

Gap: The accuracy and adequacy of IDS signatures cannot be determined.

The proprietary nature of the signatures for most commercial ID systems makes a detailed discussion of their accuracy and adequacy difficult. For the most part, signatures seem to be at a very concrete level, not far removed from simple string matching. This may be adequate for very simple attacks such as those that can abuse a CGI-bin executable with a single command, but are probably inadequate for sophisticated, multi-stage attacks.

Among the research and open source tools, the situation is somewhat different. The STAT family [R4] models attacks using abstract state machines. The machines are constructed to recognize generalizations of known attack scenarios. If the individual constructing the recognizer has sufficient insight into the attack and uses abstraction to its maximum advantage, a recognizer for a class of attacks rather than for an instance of the class may result. This happened during the 1998 Lincoln Labs evaluation when NSTAT was able to recognize a new "user to root" attack that had not been present in the training data.

Given that many of the observed attacks are the result of scripts distributed by the attack's developer, and used essentially as is by numerous "copy cat" intruders, concrete patterns have a definite role in the intrusion detection arena, just as they do in the virus arena. Were it mature, the IDS vendor community would be rushing new signatures to their users as fast as new attacks are recognized. Our worry is that the attacker community and "script kiddies" that seem to be responsible for much of the attack traffic are not the community that we should be most concerned with. There are obvious variations on most scripts that would escape detection by all commercial (and many research) ID systems. A focused attacker, targeting a system for the purpose of extracting, inserting, or modifying information is likely to be able to escape detection long enough to accomplish their goals.

These issues have a bearing on how the vendor community needs to address the release of signatures. It appears that what will distinguish the more successful ID vendor is

- the timeliness of response in releasing detection signatures when new attack mechanisms have been identified
- the quality of these signatures

In the latter case, this probably means more rigorous testing of signatures, improving the robustness of signatures to variants of the attack mechanisms (as discussed above), and

minimizing the false alarm potential of the signature. Clearly there is a conflict in addressing the two goals of timeliness and quality, but the more effective vendors will develop processes that can balance both.

3.4.8 Characterizing IDS Performance

Gap: There is no generally agreed upon figure of merit that can be used to characterize IDS performance. Acceptable performance is similarly difficult to define.

The typical IDS looks at a series of events and tries to identify those that represent an intrusion. The events may be log records from one or more monitored services on a host, packets on a network, or any other surrogate for activity within the monitored domain. Normal activities as well as intrusions may be represented by single events or by combinations of events. At the service monitoring level on a host, an activity may give rise to a series of log records (e.g., an ftp session might generate log records indicating the start of the session, successful authentication, files transferred, directories examined, and termination of the session). These records may be interspersed with the records of other simultaneous ftp sessions as well as records from other services. A monitor on the network segment serving the host would see the packets that made the session, along with all the other packets that passed along the segment during the same period. In one case, an event is a log record, in the other, it is a single packet.

When an attack occurs, it may be represented by a single event or by a sequence of related events. On the network it may be possible to accomplish the attack with a single packet, but it may also be possible to fragment that packet into a number of small pieces as discussed further in Section 3.1.5. It may require a number of related packets to carry out an attack, and again, it may be possible to fragment many of these. Similarly, a single log record may serve as a positive indication that an attack has taken place or it may require a sequence of such records, extracted from the log stream to make this determination.

We belabor this point because it is germane to the discussion of IDS accuracy and precision. From an operational standpoint, we are interested in having a system that identifies as many of the real attacks as possible while raising as few false alarms as possible. One way to characterize the former is in terms of the percentage of attacks identified by the system. Assuming that we can evaluate the system by subjecting it to known attacks, we ought to be able to compute the percentage of attacks identified. Unfortunately, for reasons that will be discussed below, even this is not easy. Ideally, we would like to subject the system to a large amount of activity that contains no attacks. The numerator of the false alarm rate is the number of attacks falsely identified; what is the denominator? If we associate an attack with an activity of some kind, we need to count activities, a problematic undertaking. Besides, there is no guarantee that the events that contributed to the false alarm all came from a single activity or even a related set of activities.

In the only set(s) of test data with which we are familiar, that used by Lincoln Labs to evaluate DARPA sponsored research systems [B29], the term "session" is used to identify related sets of packets. A session is characterized by a start time, duration, service, source and destination (IP address and port). In the test data, attacks are associated with sessions and the nature of the attack is indicated in the labeled training data. Using the session as the unit of analysis is less than satisfactory[1] since it does not provide accurate results if

- a single attack requires more than one service to be used or involves multiple sessions of the same service to be completed or
- a false alarm determination is based on data from more than one session.

Assuming that we can choose an appropriate unit of analysis (dubious) there are four possible outcomes for each unit:

1. The unit contains an attack that is recognized (true positive).
2. The unit does not contain an attack but is identified as containing one (false positive).
3. The unit contains an attack that is not recognized (false negative).
4. The unit does not contain an attack and none is recognized (true negative).

Each of these cases can be described using Bayesian or conditional probabilities. In general, we want the percentage of true positives to be high and the percentage of false positives to be low. Unfortunately, these figures alone are not sufficient to describe a system in meaningful terms since we do not know how frequent attacks are (i.e., the percentage of analysis units that contain an attack).

Unfortunately (for the purpose of the analysis), the percentage of units that represent attacks is more likely to be on the order of one in a 100,000 or so. Optimistically assuming a 100% true positive rate, we will identify every attack. Now assume that the false positive rate is 0.1%, an objective given by DARPA as a target for its research. This will result in a false alarm rate of about 100 per 100,000 and the staff will have to examine 101 analysis units to find the one true intrusion. Recent papers by Axelsson [R77] explore this in detail.

Perhaps the most meaningful figure of merit for a deployed IDS is the ratio of the true positive rate to the false positive rate. Some experimentation is probably required to determine the range of acceptable ratios, but it is unlikely that systems generating a preponderance of false alarms will be acceptable. If we consider 5 false alarms for every real alarm to be the limit of acceptability, a system that identifies one real attack in 100,000 units of analysis must raise false alarms in no more than 5 cases per 100,000 or 0.005%, a far more stringent requirement

1. The session concept was used in the 1998 evaluation. In 1999, the session concept has been dropped and false alarms are characterized per unit time. While this avoids the problem of choosing an appropriate unit of analysis, it results in a performance figure that is a function of both the environment and the system under test.

than that imposed by DARPA. As the attack frequency falls, the false alarm requirement becomes more stringent. Translating these requirements into more easily measured base quantities such as packets or audit log records requires an understanding of the detection process and how it forms its analysis units.

The Lincoln Lab's evaluation of DARPA research systems has stated that the acceptable limit on false alarms is about 100 per day, arguing that this is the maximum number of intrusions with which a single analyst can deal. Even if an analyst can sustain this rate of intrusion handling, we doubt that the analyst will be able to sustain the level of performance necessary to identify the single real attack per day that is hidden in the 100 or so false alarms predicted by a 0.1% false alarm rate.

In summary, both the Bayesian detection rate and the Bayesian false detection rate, along with the traffic level and the attack rate, are factors in determining whether the performance of an IDS is acceptable. If the number of false alarms per day is acceptable, but the attack rate is too low for reliable discrimination by system staff, it could be desirable to add additional (preferably benign) intrusions into the analysis stream to provide both analyst stimulation and system calibration.

3.4.9 Real Network Traffic Is Not Well Behaved

Gap: The false alarm problem is exacerbated by the fact that legitimate traffic often contains the kinds of pathologies typically associated with attacks.

The more theoretical arguments of the previous section are reinforced by Paxton [R84] who believes that "...the whole concept of [network] intrusion detection is doomed to failure. Attacks on the monitor, such as overloading it with too much traffic or using software faults to bring the monitor down can be defended against, but that still leaves the problem of 'crud' that looks like an attack but isn't one." Some odd-looking but legitimate traffic includes

- storms of FIN and RST packets
- fragmented packets with the "don't fragment packet" flag set
- legitimate tiny fragments
- data that is different when retransmitted

Paxton believes that the in inability to discriminate between "crud" and malicious packets will be exploited by attackers. For these reasons he feels that host-based ID systems are likely to predominate.

The resulting discussion leads to the dismal conclusion that false alarm rates in current network-based systems may be so high that system administrators will ignore all warnings.

We believe this should be a wake-up call to the research community. Research efforts in the intrusion detection field have focused increasingly on building ID frameworks that address higher level issues and abstractions. We believe that these efforts should, in great part, be redirected to address the fundamental issues of minimizing false alarms and quantitative testing of the resulting algorithms. If the false alarm issue is not resolved, all the frameworks in the world will not help.

3.5 Data Analysis Needs

3.5.1 Testing ID Systems

Gap: Very little has been done to objectively evaluate and test ID systems.

Section 3.4.8 briefly discussed research systems testing. The misconceptions about commercial systems are even worse. Reading vendor literature gives the impression that commercial products currently provide robust defense against intruders. However, independent IDS evaluations (including the evaluations performed for this study) indicate that this is not the case. Generally vendors neither provide information about the performance of their products, nor do they make their detection algorithms public. The latter point is motivated by the need to hide attack signatures from competitors, and the need to prevent intruders from knowing exactly how the signatures work. A third, but not readily admitted, motive may be that by publicizing signatures, it will become apparent how simplistic many signatures really are. Full public scrutiny of attack signatures may, in the short term, result in an increased number of intrusions, but in the long term the scrutiny would motivate the development of more robust approaches to intrusion detection. In the meantime, black-box testing is the only public means to determine the diagnostic accuracy of these tools.

Testing is difficult since an IDS cannot realistically be examined in isolation but needs a network, or a network simulator, with which to interact. Several such environments are being developed within the research community [B7, B9, B16, B38] and we hope they will migrate into use within the larger user community. Testing also requires significant volumes of complex data (either captured from a real network or developed synthetically) against which the IDS can be evaluated [R25, R92, R93]. In addition, many intrusions depend on one operating system or even the specific release of an operating system. Thus test data may need to be finely tuned to the operating environment.

However, "...some of the problems that we discovered were so basic (the conditions leading to these problems occur frequently even in normal traffic) that it appears as if no in-depth testing has been done at all", and "the most important issue that vendors need to address is testing" [B26-b].

Independent tests have been performed on commercial products (see Appendix D for details). However, given the immaturity of the field, the rapid evolution of existing products, and the frequent introduction of new products, these tests are often obsolete as soon as they appear. New attacks and variants of old attacks occur frequently, and test data can soon become outdated. Any useful testing methodology must address the issue of obsolescence. It must also address the general testing issues discussed above.

Penetration testing can determine whether an IDS system is functioning properly, as well as evaluating system or network security. This is performed by making "friendly" attacks against the network and hosts (and their ID systems) as opposed to testing an IDS prior to installation. While more reactive, this type of testing does not require one to develop large databases of test data. Human actors serve in the role of attackers, directly using the tools and techniques that attackers use. Penetration testing is discussed in the papers "Improving the Security of Your Site by Breaking into It" [B107], and "Broadening the Scope of Penetration Testing Techniques" [B105]. While penetration testing is an important assessment tool, it may only tell you that the IDS you have purchased is inadequate.

The testing of ID systems is difficult. Given the spurious results that most commercial ID systems too often generate, it is understandable why vendors are reluctant to be involved in public testing. Several organizations, mostly research oriented, are developing environments for evaluating IDS tools, but as these environments are themselves immature, they have not seen significant use in evaluating commercial products. Over the long term, IDS testing environments need to generate test data on the fly to address the growing number of new attacks.

3.5.2 Reduction and Analysis of Audit Data

Gap: Tools to effectively manage audit data are lacking.

For complex networks, storing raw real-time audit data for retrospective ID analysis may require impractically large storage capacities. Thus, there is a need to develop techniques that allow compression of the data so that loss of information, important to subsequent ID analysis, is minimized. Issues that need to be addressed in this area are

- What data should be collected and when should it be collected?
  The answer to this depends on the audit needs and is constrained by whether the IDS is network- or host-based. On the network side, is packet header information sufficient or is packet content important? If content is important, the storage problem is much more severe. Audit data storage needs could be reduced if the data is only sampled (this might also help with performance) but what is the basis for sampling? Since storage capacity is not infinite, there must also be mechanisms to discard or de-emphasized older data.

- What approaches to data reduction should be used [S2, R68-a]?
  The most direct solution is to use data compression techniques [B70]. This approach usually allows the original data to be completely restored. However, if loss of "unimportant" data can be accepted, then greater compression can be achieved by filtering or using learning techniques originating from the artificial intelligence community. Techniques such as rule induction, neutral nets, and genetic algorithms all provide means to compress data into forms that require much less storage. These AI approaches tend to generalize over that data, removing irregularities as they do so. As a result of this data abstraction, they may also provide insights into patterns in the data. However, full restoration of the data is not possible, and this may have implications for forensic analysis. These learning algorithms are discussed in more detail in Section 3.6, Advanced Research.

- What is the performance impact of different compression strategies?
  Performance is affected by computational and communication overhead and there is likely to be a direct trade-off between these. Network and some host ID systems consist of a set of sensors and a central management unit. If the sensors transfer all the raw audit data to the central unit for storage, then the communication overhead will be high, and this may affect the overall performance of the network. Some of this overhead can, however, be reduced if the sensor units perform local data reduction — with a resulting load on the local CPUs. Trade-off analysis, specific to the network, would have to be performed to determine the optimum balance between CPU and network traffic loads. This issue represents a "mini-gap." A different performance issue results from the use of learning algorithms. Given the large quantities of data that must be managed, such algorithms must be capable of incremental learning, but many are not.

- What issues arise in integrating diverse data sources [S14, R45]?
  Compression using learning techniques will generally require a degree of regularity in the data structures. This implies that, when raw data from different sources are involved (e.g., host vs. network data or data from two different types of IDS), the data will need to be integrated prior to compression. Integration may have performance implications and the practicality of integration will depend on the effectiveness of data exchange formats [R68-b]. The consistency and completeness of data from different sources may also be problematic. Finally, when integrating diverse data sources, there could be difficulty in determining the relative timing between logged events due to the use of different clocks.

With the ever-increasing size and complexity of networks, managing and storing audit data will become an increasingly challenging but important problem — not only to the IDS community, but to network management in general (e.g., for performance analysis). While data compression and filtering technology currently exists, its application to audit record reduction is still in its infancy. As mentioned in Section 3.4.3, it is important, especially for security-critical applications, to store audit records in a physically secure device (to prevent insider corruption) on write-once media (to prevent outsider corruption).

3.5.3 Forensic Analysis

Gap: ID systems could provide evidence to support prosecution but do not.

With the rapidly growing theft and unauthorized destruction of computer-based information [B51], the frequency of prosecution is rising. To support prosecution, electronic data must be captured and managed in such a way that it provides legally acceptable evidence [B50, B58, B77-a]. The discipline of computer forensics, which lies at the intersection of electronic information systems and the law, is still maturing. To date, very little has been written about the role of ID systems in forensic evidence collection. However, ID systems have the potential to provide significant capability in gathering forensic evidence. They can support time-lining of suspicious activities such as network scans, login attempts, and document modification.

Permanently recording the vast quantity of information flowing across a large network is, in practice, impossible. However, if a network IDS has identified a signature suggesting an intrusion, it could selectively record the relevant packets for subsequent legal examination. Thus, an IDS could greatly reduce the volume of data needed for legal support. Unfortunately, given the false alarm rates of current network ID systems, it may be difficult to make a strong case for evidence collected this way. In some cases evidence might simply be missed, and in other cases much irrelevant information might be collected. Whether such information would hold up in a court of law remains to be seen.

A more promising approach in the short-run is described in a paper by Marcus Ranum [B68]. Much of what he suggests can be implemented using a variety of existing system administration tools such as NNstat [R69] or tcpdump [R37]. Ranum believes that "...it's almost always going to be a manual process. If I developed a tool that did these things automatically, then the bad guy would say 'Oh, well I'm going to hide it in one of the places he's not looking.'" ID systems tools such as Tripwire [C22], that provide file integrity checking, can be of benefit. An image of the system parameters is captured and stored in a secure database prior to system operation. This is subsequently used to test for unauthorized changes such as modified files, password revisions, and changed network configurations. These configuration changes may be used as evidence.

Most computer forensic evidence focuses on the contents of files, emails, and other artifacts. While in theory, commercial ID systems generate intrusion-related information, a major problem with their use is that of high false alarm rates. Without more accurate and consistent analyses of the raw data, today's ID systems produce output that would likely lack credibility in the courtroom. At best, these systems might help point to evidence that can be analyzed and justified manually. Some research-oriented ID systems [R30] produce process-related evidence (i.e., information pertaining to sequences of activities that the intruder performs).

This may be of considerable future use in legal settings. It appears that the current immature state of ID systems will limit their use in legal battles, but with improved accuracy, their potential for future use is possible.

3.6 Advanced Research

The first three subsections deal with machine learning algorithms: example classification, neural nets, and genetic algorithms [B84] that can be applied to intrusion detection. While they have many characteristics in common, these methods are discussed separately as each has unique characteristics.

These methods all generate rules from data, but the rules are not designed for human understanding. In general, example classification produces rules that are most easily interpreted by humans, while neural nets generate the most opaque rules. However, rule clarity is not a strong feature of any of the machine learning methods. These systems are no better than systems that attempt to recognize a priori patterns associated with attacks or intrusions. They are of research interest because they may support the recognition of intrusions that have not been seen previously and which have no previously described patterns.

As discussed in each of the sections below, there are many issues to be resolved before machine learning techniques can be used in operational environments. Despite these challenges, machine learning techniques have real promise since they are usually

- computationally efficient
- require relatively little storage
- can adapt to new data

It is too early to determine which of these approaches, if any, will dominate, although greatest progress to date appears to be with computer immunology, a type of example classification.

In addition to the machine learning approaches, we consider several other advanced research areas that merit further investigation. The state/Petri net approaches have been explored in some detail while data fusion and graphical representation have not.

3.6.1 Learning Through Example Classification

Gap 1: Current ID systems do not generally recognize new types of intrusion.

Gap 2: Current systems depend on subjective human interpretation of intrusive behavior.

Learning through classification as applied to intrusion detection is a technique that generalizes from examples of data derived from user or system activity. In some classification approaches only examples of normal activity are employed, and anything outside this regime is considered anomalous, while in other approaches examples of specific intrusions may be learned. Because the emphasis of this approach is on characterizing normal behavior, it may be considered a branch of anomaly detection. (See section 3.4.5)

One approach to classification learning is computer immunology, inspired by the mechanisms of biological immunology [R47, R48, R49, R50, R67]. Forrest, et al. argue that computer networks will always be vulnerable to intrusion since security policies, computer programs, and system configurations will never be fault-free, and that intrusion detection must live with these realities. Vulnerabilities in biological systems are overcome by immune systems that are distributed, diverse, autonomous, and self-repairing in their approach to dealing with foreign bodies. Biological immune systems recognize foreign bodies whose characteristics have not been seen before [R49].

Motivated by these ideas, computer immunology provides a basis for learning from examples of behavior that is considered normal, and identifies as suspicious, sequences of behavior that do not conform to its notions of normalcy. Initial experiments were based on training the system to recognize short sequences of legitimate system calls that result from running programs such as sendmail. These experiments found that "...short sequences of system calls define a stable signature that can detect some common sources of anomalous behavior in sendmail and lpr" [R48].

When the author repeated the experiment with data generated at MIT, similar results were obtained, but the pattern set describing normal was very different [B87]. Similar results have been obtained for patterns of network activity and for program instruction sequences. This approach is sensitive to both changes in the program and changes in the usage pattern of the program that affect its observed signature. Stillerman, Marceau, and Stillman [R60] are using computer immunology to address the issue of intrusion detection with respect to distributed object technology (specifically to CORBA applications), while Wespi, Dacier, and Debar [R91] are investigating the use of variable length patterns to determine if these patterns are both more accurate and more concise.

Although not motivated by immunological principles, two other research efforts also use short system call sequences to identify normal and anomalous behavior. Lee, Stolfo, and Chan [R17] generate rules inductively [R19] using the data defined by Forrest [R48] but augmented with sequences that reflect intrusive behavior. The generated rules can be adjusted either so that they detect normal behavior (thus classifying all other responses as potentially intrusive in the "anomaly" sense) or so that they detect specific intrusions (in the "signature" sense). They suggest that "...while the results here show that our approach generated much 'stronger signals' of anomalies from the intrusion traces [than did Forrest's results], it should be noted

that the method in (Forrest et al. 1996) did not use any abnormal sequences from the intrusion traces in training." This work of Lee, Stolfo, and Chan is part of a larger effort called the JAM (Java Agents for Meta-Learning) project [R15, R16, R53]. Work by Helmer, et al. [R62] is tackling the issue in a similar manner, namely using Forrest's data to inductively generate rules for anomaly detection. The focus of their work is on the use of distributed agents whose logic is based on these rules.

Research in this area is important because it overcomes weaknesses in more conventional systems. These approaches do not exclusively recognize known intrusions (although the work of Lee, Stolfo, and Chan can also identify known intrusions). Systems based on these techniques are also capable of learning so that they adapt to changing circumstances. In principle, learning techniques can also be sensitive to what the data is saying directly rather than to human interpretation of that data. In this sense, diagnosis is more grounded in the empirical data.

Still, significant issues remain:

- These methods generally require much fine-tuning to operate at maximum efficiency [R49], but the optimum parameter values are not always obvious.
- The learning process may be computationally expensive, particularly if it cannot be performed in an incremental way. An associated issue is that one may need to "unlearn" old experience. This introduces complexity in retraining.
- A large amount of data is required to train these systems. The difficulty of providing data that is guaranteed not to contain attacks or attack-like behaviors is problematic. This is complicated by the fact that "normal" is a function of both time and place.

3.6.2 Learning Through Neural Nets

Gap: ID systems need to demonstrate greater adaptability to new threats.

Some early experiments have been performed to determine the effectiveness of neural nets in intrusion detection. As with the classification approach, these experiments are exploratory, and their ability to scale up to meet operational needs remains to be seen. Several groups have explored this approach [R46-b, R61, R70].

Bonifácio, et al. [R46-b] and Cannady [R70] both train back propagation neural nets to recognize sets of known network attacks. Within the scope of the experiments, the results were able to identify intrusive patterns.

However, the following questions remain to be resolved:

- While relatively small examples sets may produce acceptable diagnoses, large example sets could crowd the solution space. This could result in unacceptable error rates or even result in non-convergence of the solution.
- Even if the solution does converge, training times for large data sets may be unacceptably high.[1]
- An inherent problem with neural nets is that they provide little or no insight into the logic that drives the conclusions they generate.
- Neural nets generally require fine-tuning of network topology and also of the algorithm's parameters. This tuning can be as much art as science.
- The work in both of these papers depends only on packet header information. This limits the scope of intrusion types that can be diagnosed.

A problem with the above approaches is that the neural nets were trained to detect known attack signatures. In "Detecting Anomalous and Unknown Intrusions Against Programs" [R61], Ghosh, Wanken, and Charron address the issue of training a back propagation net to identify unknown attacks. Their basis is similar to that used in the above classification approach, namely, training the net on a range of normal behaviors and detecting any patterns that deviated from these behaviors. They also added random patterns (which were considered anomalous conditions) to the "normal" training set, and this significantly reduced the frequency of false negative diagnosis. However, the scope of the experiment was quite limited and focused only on use weaknesses in the Linux lpr program. While the results were encouraging, it remains to be seen if the approach can be scaled up.

To date, neural network approaches have demonstrated the ability to diagnose attack patterns. However, issues of diagnostic accuracy, training times, solution convergence, fine tuning, and opaqueness of the diagnostic logic need to be further addressed for large real-world systems. On the positive side, neural networks work well with noisy data, require less labor intensive model building, and perform diagnostic computations very rapidly — thus making real-time diagnosis possible. In addition, neural nets can be trained from operational data — they do not need human insights that may possibly be misguided. Early results [R61] indicate that the technology can detect unknown intrusion patterns, so this line of research is worthy of further study. Other neural networks architectures such as self-organizing feature maps [R70, R75] have been proposed and may overcome some of the limitations of the back propagation approach.

---

1. In Cannady's preliminary work, it took 26 hours to train on 9462 examples. He used output generated from the RealSecure™ IDS with simulated input (e.g. denial of service and port scans) from Internet Scanner™. Given the high false positive rates that RealSecure can produce, there is also some question about the neural net's diagnostic accuracy.

There is one application of neural nets to intrusion detection that radically differs from all others. Just as individuals have distinct handwriting signatures, each person may have a unique dynamic typing "signature" that reflects their idiosyncratic use of the keyboard. It may therefore be possible to train neural nets to recognize personal typing signatures, and by implication to identify intruders [R42, S5]. The use of typing signatures has the significant advantages over other anomaly approaches in that it is stable over time (independent of changing applications etc.) and is very difficult to forge. However, this approach will only be applicable to insider threats if the insider makes unauthorized attempts to access information. This technology has been applied commercially to the recognition of passwords [C14], but its use in the broader arena of intrusion detection is worth exploring.

3.6.3 Learning Through Genetic Algorithms

Gap: ID systems need to demonstrate greater adaptability to new threats.

Applying the genetic algorithm approach to intrusion detection, string sequences are used to define the instructions for diagnostic tests [B85, B84]. These instructions, which test for normal or abnormal behavior, are evolved during an initial training period to improve the strings' diagnostic ability. Prior to training, strings have little diagnostic power, but by randomly recombining segments of these strings, new strings are created, and the fittest of these (from a diagnostic perspective) are selected. This cycle of recombination, testing, and selection is continued until improvement ceases. At this point the strings are used in the operational environment.

Some limited work has been performed using the generic algorithm [R22, R27], but the approach remains to be proved. In "Active Defense of a Computer System Using Autonomous Agents" [R22], the algorithms were trained on a small set of known attack patterns (port flooding, port walking, probing, and password cracking). Brief results on limited data indicated some weak ability to discriminate attacks from normal behavior.

Genetic algorithms still have to prove themselves in the ID arena. Many of the issues discussed with respect to neural nets are relevant to genetic algorithms. However, one advantage they do have over neural nets is that the diagnostic rules they generate are easier for people to understand.

3.6.4 Data Fusion

Gap: ID systems often produce inaccurate diagnoses resulting from insufficient information and simplistic analysis.

Most current commercial network-based ID systems rely primarily on a signature matching using IP header information. Even those that support both network- and host-based intrusion

detection do not correlate the separate diagnoses to any great extent. This lack of cross-checking contributes to the high number of false alarms. In addition, future attacks, for example distributed or slow attacks, will increasingly stress current detection capabilities.

This suggests that the integration of multiple sources of information supported by diverse types of analysis (i.e., data fusion) should be an important research focus.

There has been significant activity in the data fusion community in recent years, both in the government and private sectors [B145]. The DoD has extensively supported research and has helped the field to mature. These efforts have led to the unifying of many diverse technologies and methods in disciplines such as statistical estimation, digital signal processing, control theory, and artificial intelligence. This foundation will significantly help in supporting data fusion within the intrusion detection area.

While the need for data fusion has been recognized for some time, not much has been implemented. Quoting Lunt from a 1993 paper [R1-a], "...in order to effectively address various intrusion threats, a system should combine several intrusion-detection approaches. We should begin to see intrusion detection systems that can intelligently make use of audit data gathered at several levels from the monitored system (e.g., system level calls, command line level, and application level). Profiling files and programs will give another dimension along which to characterize expected behavior on a system." More broadly, we may need to integrate diagnoses from multiple local hosts with network-based diagnosis.

In addition, it may be necessary to coordinate these diagnoses with diagnoses from other administrative domains. Thus, diagnosis is needed at multiple levels of abstraction [R52-a, R2-a]. This view is reinforced by the paper "A Glimpse into the Future of ID" [R6-c]. Here it is argued that, because of the voluminous quantities of data that need to be processed by an IDS, local distributed processing is essential in order to save on communication overhead. This paper predicts that fusion-based ID systems will need to synthesize data (from packet sniffers, log files, etc.) related to ongoing intrusions with longer term information about the system, histories of past intrusions, user profiles, etc. Fusion-based ID systems will also need to know spatial-temporal aspects of the data and the network.

The ICE project has undertaken an ambitious effort to address data fusion issues [R51]. This tool will provide support for audit trail analysis, statistical anomaly detection, model-based reasoning, and several other approaches. It uses Bayesian inference techniques [R72, B84] to integrate the results of the individual analyses.

The matter of how to combine data probabilistically is central to any fusion technique, and there are a variety of techniques, including the Bayesian approach [R73], that do this. Some data is more credible than other data, while the results of one analysis technique may be more reliable than another. This information should be considered in reaching a confidence level

associated with the diagnosis. Because of the complexity of the analysis, it is also desirable to integrate human reasoning (see Section 3.3.2) into the decision-making process and to support this reasoning with data visualization (see Section 3.6.6). However, to exploit these capabilities, multiple levels of information abstraction need to be supported.

Currently commercial systems are weak because they provide no indication about the confidence of the diagnosis, and they perform analysis only at the lowest level of abstraction (i.e., on data patterns).

Integrating information from diverse sources is critical in reducing high false alarm rates (see Section 3.6.4). The high rates of today's commercial systems will not be tolerated indefinitely, and long term success of the intrusion detection business depends on honestly confronting this problem. We believe that part of the solution is to apply the techniques discussed in this section.

3.6.5 State Transition and Petri Net Modeling

Gap: ID systems need to improve diagnostic accuracy.

State transition diagrams (represented as networks of nodes and arcs between the nodes) can be used to describe states that a system can be in (the nodes) and the possible transitions between these states (the arcs). In the context of intrusion detection, this representation can be used to describe the sequence of actions (the arcs) that an intruder must take to move from some legal state (the initial node) to a final compromised state (the terminal node). Several research efforts have evaluated this approach

According to Kemmerer, et. al. [R4], "Unlike comparable analysis tools that pattern match sequences of audit records to the expected audit trails of known penetrations, STAT rules focus on the effects that the individual steps of a penetration have on the state of the computer system." It is claimed that this approach is more robust in detecting unknown vulnerabilities, since it emphasizes the compromise of defined system states rather than focusing on audit-trail symptoms of an unknown intrusion. One potential weakness of the implementation described by Kemmerer is that it is too deterministic in defining state transition sequences. There can be variability in the order in which some partial sequences of the intrusion can be composed to build the full sequence, and this variability is not accounted for. Further information on the STAT series of tools can be found in Section 2.1.1.2. Similar research has also been performed by groups at Purdue University [R7] and University of Idaho [R65]. In both cases the Petri net formalism is used to express state transitions, and partial sequences are admitted.

While in theory these efforts may provide more accurate diagnosis, there has been insufficient testing to demonstrate this. From a performance standpoint, preliminary data [R5] indicates that the overhead associated with running a state-based IDS is significant and may not support

real-time operation. However, it should be kept in mind that these efforts are research oriented, and minimizing performance impact is unlikely to be a high priority. This work has potential in that it operates at a level of abstraction above that of raw audit data. In doing so, it exploits notions of system state and temporal flow of events, and consequently may provide a means to reduce false alarm rates while also detecting novel attacks.

Also, because of the relatively high levels of abstraction involved, this approach may someday allow a reusable knowledge base to be "ported" to detect intrusions on different kinds of machines, networks, and applications.

3.6.6 Graphical Representation

Gap: ID systems lack support for human analysis of intrusion-related data.

A visual presentation can help people interpret large quantities of data. Within the intrusion detection field too little attention has been given to this area. Most ID systems do not provide any way of viewing information other than through lists, aggregates, or trends of raw data. Ideally, however, people would be able to compute arbitrary functions on host and network data, to graphically view the functions using multiple visual formats, and to update the displays in real time so as to track events. Such capability could provide security managers with earlier warning indicators of an attack, provide additional assurance that machine diagnosis was accurate, and might provide insights indicating attacks of unknown types.

Vert, Frinke, and McConnell [R64] have investigated one approach to visual presentation. They use "spicules," spherical objects with embedded radial vectors to graphically portray information. These vectors, one for each host, are associated with the number of active system processes, the number of active user processes, the percent of used system resources, and the number of open system files. As the values of these variables increase, the vectors rotate to the vertical. Multiple spicules can be embedded in a three-dimensional space, and the thickness of the links between two spicules indicates the degree of communication between them. However, this research is still formative and the field in general needs to be explored in much greater depth. Graphical representation has potential to support computer human interaction as discussed in Section 3.3.2.

4 What Are the Organizational Issues?

It is as important to have effective security policies in place as it is to have the latest technology if one wants to secure a network against intrusions. Effective policies will help to ensure that, among other things, security managers are adequately trained, information derived from ID systems is interpreted correctly, and appropriate actions are taken when an intrusion is identified. These issues are particularly important since, unlike virus detection systems, ID systems require care and attention for successful operation.

4.1 Barriers to Effective Security

There are many reasons why organizations fail to provide effective security. Recognizing these may help an organization develop more effective security strategies. Recent surveys summarize some of the reasons that organizations give for their failure to act in this area. The ICSA/SAIC 1999 security survey (745 respondents) [S33] posed the question, "What is the SINGLE greatest obstacle to achieving adequate information security at your organization?" Replies fell into the following categories.

Obstacle	Percentage of organizations
Budget constraints	29
Lack of senior mgmt support	14
Lack of employee training/end user awareness	10
Lack of competent IS personnel	9
Lack of internal policies	8
Lack of centralized authority	8
Technical complexity	6
Unclear responsibilities	4
Lack of good security products	3
Other	9

TABLE 4-1: BARRIERS TO ID SYSTEM ADOPTION - 1

The *InformationWeek* 1999 global security survey (2,700 respondents in 49 countries) [S30] states the following in response to the question, "Which of these is the most significant barrier to effective security in your company?"

Barrier	Percentage of organizations
Lack of time	17
Complexity of technology	16
Pace of change	11
Lack of management support	11
Poorly defined policy	10
Capital expense	8
Lack of dept/group cooperation	8
Lack of training	8
Lack of qualified staff	6
Labor expense	5

TABLE 4-2: BARRIERS TO ID SYSTEM ADOPTION - 2

A survey [S36] of 1850 computer-security experts and managers taken by the SANS Institute at conferences held in May 1999 identified "Seven Top Management Errors that Lead to Computer Security Vulnerabilities." They are

- *Pretend the problem will go away if they ignore it.*
- *Authorize reactive, short-term fixes so problems re-emerge rapidly.*
- *Fail to realize how much money their information and organizational reputations are worth.*
- *Rely primarily on a firewall.*
- *Fail to deal with the operational aspects of security: make a few fixes and then not allow the follow through necessary to ensure the problems stay fixed.*
- *Fail to understand the relationship of information security to the business problem — they understand physical security but do not see the consequences of poor information security.*
- *Assign untrained people to maintain security and provide neither the training nor the time to make it possible to do the job.*

While these responses address the broader topic of information security, they are clearly germane to a specific security technology such as intrusion detection systems. This section describes several of these issues as they relate to the use of ID technology including

- understanding the threat as a prerequisite for determining effective means for protection
- the critical role of management sponsorship and support
- setting the appropriate policies, procedures, and mechanisms for their enforcement
- understanding the full IDS life cycle and what is required to support it
- heightening awareness and regular training
- factors to consider in the decision to make, rent, or buy ID staff capability
- managing expectations of what ID technology can and cannot do

4.2 Understanding the Threat

Before an organization makes an investment in security technologies, it is important that it understand what assets require protection as well as the real and perceived threats to those assets. Threats can be characterized by the type of attack, the category of the attacker in terms of their capabilities, resources, and goals, and the organization's tolerance for loss of the asset that is being protected. Loss can be characterized as a loss of confidentiality, availability, or integrity.

An attacker can have one or more objectives in attacking a computer network. Several of these objectives are identified in Table 4-3, which provides an approximate correlation between types of attacks and the objectives of attackers. In the table, information modification connotes the clandestine change of data (i.e., so that the changes are not noticed) while information corruption renders the information unintelligible.

These modes of attack may reflect different attack signatures; this may imply different intrusion detection strategies. For example, information retrieval is likely to be performed using a stealthy attack, while for information corruption, stealth may be less important than speed. A bank, concerned with illegal financial withdrawals, likely considers detecting stealthy attacks very important.

Hence, the chosen intrusion detection approach depends, to some extent, on the objectives of an organization's likely adversaries. A paper titled "Who's stealing your information?" [B51] describes who is involved in illegal information retrieval, a current problem causing a great deal of financial loss, and likely to become much worse. The *InformationWeek* survey [S30] suggests that the problem of insider attack is perceived to be decreasing relative to the problem of attack from external intruders and terrorists.

Determining whether the potential attacker is inside or outside of the organization's infrastructure has a bearing on the type of IDS you select.

Objective	Denial of service (loss of availability)	Information retrieval (loss of confidentiality)	Information modification or corruption (loss of integrity)
Curiosity		X	
Vandalism	X		X
Revenge	X		X
Financial gain			X
Competitive advantage	X	X	X
Intelligence gathering		X	
Military gain	X	X	X

TABLE 4-3: INTRUDER MOTIVES

4.3 Management Sponsorship and Support

Often the most significant obstacle to the success of an information security improvement initiative is lack of management support[1]. The information in Tables 4-1 and 4-2 is consistent with our experience at the Software Engineering Institute in implementing improvement initiatives, including those focused on security improvement. Managers have many goals to meet, and they must often make compromises between them. In today's competitive environment, companies are focused on achieving market share and minimizing product cycle time, achieving financial and other performance objectives, and worrying about mergers, acquisitions and reorganizations. In such a climate it is no surprise that security concerns are often relegated to a lower priority. Only when some significant security breach has occurred that affects a high priority business objective (such as preserving the organization's reputation) is the importance of security elevated.

Management sponsorship is demonstrated by

- visibly being supportive of efforts to improve information security
- encouraging staff to communicate security concerns at all levels of the organization

1. One individual told an author of this report that he obtained management sponsorship by demonstrating how easy it was to break into his manager's confidential computer files. This approach is not necessarily recommended, but at least in this case, appears to have been effective!

- follow-through on the concerns that are expressed
- the sustained allocation of sufficient resources to accomplish security improvement initiatives (budget, staff, time)
- the existence of an organizational security policy and procedures that are documented, understandable, not open to misinterpretation, not overly burdensome or restrictive, and that cover the range of topics required to meet the organization's security objectives
- visible, demonstrated enforcement of the policy including consistent application of sanctions for non-compliance

Unless there is significant Chief Information Officer and Information Security manager-level sponsorship and support for the deployment of an IDS and all that it implies, the successful operation and use of this technology will be short-lived, sustained only by the interest of those internal champions who believe in its benefit, until the next high-priority item requires their attention.

4.4 Policies, Procedures, and Mechanisms for Their Enforcement

Establishing an information security policy so that it reflects an organization's business goals and enacting that policy so that it is a normal, accepted part of day-to-day operations are difficult at best. The business goals need to be well articulated both initially and as they change. The security implications of each business goal need to be identified and transformed into organizational security requirements. Meeting these requirements needs to be folded into the objectives and incentives of the responsible managers. Once the requirements are well understood, a meaningful security policy (and supporting procedures) can be developed with reasonable confidence that the content and intent will be enacted.

Charles Cresson-Wood [B131] provides a comprehensive description of how to go about developing an information security policy and of the content items that should be considered. There are numerous additional references that address this topic. For more specifics on policies related to detecting and responding to intrusions, refer to the SEI reports *Detecting Signs of Intrusion* [B98], *Preparing to Detect Signs of Intrusion* [B116], and *Responding to Intrusions* [B123]. Developing the content is the easy part; making it real in the behaviors and actions of the organization's managers and staff is more difficult.

Effective means for security policy deployment include

- disseminating the security policy to all employees and requiring their agreement to follow it, demonstrated by their signing off on the policy
- providing policy information during security awareness training. Initial training should be given to new employees and periodic refreshers should be given to all staff, including

managers. The fact that senior managers and executives are willing to make the time to attend such training sends a strong message of support and importance (see Section 4.5).

- visibly enforcing the policy, including consistent application of sanctions for non-compliance
- regularly scheduling security policy topics during management reviews and staff meetings. Security policy reviews should identify and update areas that are not working well, reflect new business directions, and examine policy effectiveness in meeting organizational security requirements and their corresponding business goals
- involving all stakeholders who are affected by any security policy development or update. Even though this takes more time and resources up front, it saves significantly in deployment by achieving the necessary buy-in and commitment required to ensure success.

4.5 The IDS Life Cycle

An organization needs to fully appreciate the commitment required before deploying an ID system. Otherwise, the project runs the risk of wasting time, money, and staff resources in the initial phases of the life cycle. Sufficient resources must be committed at a level appropriate to cover all phases of IDS use. The life cycle phases that are described in this section include

- deploying a defense-in-depth or layered security architecture
- evaluation and selection
- deployment
- operation and use
- maintenance

Defense in Depth

The following steps should be taken whether or not the decision is made to deploy an IDS:

- eliminating as many known vulnerabilities as possible (applying patches, securing or hardening configurations)
- deleting unnecessary services
- using one or more firewalls for limiting access (both externally and internally, if required)
- implementing access control and user authentication mechanisms such as encrypted or one-time passwords, smart cards, and access control mechanisms at the network, system, application, and file levels
- using internal and external monitoring tools to detect suspicious events

- using WORM (write once, read many) or append-only log devices and files to prevent intruders from covering their tracks by deleting log records
- managing and monitoring modem connectivity to ensure unauthorized modems are not being used to circumvent firewalls
- verifying configurations through systematic "self-attack" or penetration testing
- using an integrity checking tool such as Tripwire [C22] to detect policy non-compliance
- using vulnerability scanning tools and proactive system administration for reducing vulnerabilities
- using virus detection and eradication software
- conducting on-going end user awareness training on operating securely, what constitutes suspicious behavior, and how to report it
- conducting on-going security training for system and network administrators

Evaluation and Selection

This phase of the IDS life cycle first involves determining if one will install and manage the IDS in-house or involve outside agents in these activities. Section 4.7 provides some pros and cons with regards to making this decision. If one decides to perform the implementation in-house, then selecting the IDS that best satisfies an organization's requirements and goals is essential. This involves collecting and analyzing all relevant vendor information, asking hard questions, potentially performing IDS testing, and making an informed selection decision.

A summary of topics to consider when evaluating ID systems includes

- appropriate approaches to intrusion detection
  - host and/or network capabilities
  - signature and/or anomaly-based approaches
- detection and response characteristics
  - accuracy of ID system's diagnosis (frequency of false alarms)
  - ability to customize signatures
  - proprietariness of signatures
  - ability to suppress responses from signatures that exhibit high false alarm rates
  - ability to give confidence level with diagnosis
  - providing guidance in response to attack
- performance

- speed of attack detection
- robustness of IDS under attacks on itself

- ease of use
  - effectiveness of user interface
  - ease of installation
  - the need for specialized hardware for operation
- non-technical issues
  - cost of system, including base software, installation and operation
  - company reputation (including stability, longevity, responsiveness)
  - effectiveness of documentation and training
  - rapidity of signature update from vendor

More detailed information that expands each of these points can be found in the paper "Tough Questions for IDS Vendors" [B14] and in Appendix F. This appendix contains excerpts from a paper by Amoroso and Kwapniewski titled "A Selection Criteria for Intrusion Detection Systems" [B57].

Deployment

Prior to operational use, there are a number of IDS topics that need to be considered. These include

- locating the ID sensors to protect those network resources where the most valuable assets reside, rather than trying to protect every resource on the network
  - requires ranking the value of assets according to their priority before sensor placement
- installing and configuring the IDS to reflect the security policies (Section 4.3)
- establishing ID policies with supporting procedures that
  - define attack signatures and anomaly-based profiles
  - identify procedures to collect and analyze intrusion data, and respond to the intrusion (see Section 5.2.2).
  - specify conditions under which automated response is permitted and how the outcome of such a response is monitored to ensure the appropriate action is taken
  - provide guidance on when to escalate attack information to management
  - guide the collection forensic evidence
  - address legal issues

- possibly installing a honeypot [C21-3]
- establishing initial anomaly-based profiles
- establishing initial attack signatures (whether provided by the vendor, adapted from the vendor's signatures, or developed in-house)
- accounting for the fact that ID systems cannot typically report on application-generated security events as they are generally monitoring only operating system or network events. Certain application events where knowledge is available (e.g., http attacks) can be included.

Operation and Use

In the Operation and Use phase, consider processes, procedures, and mechanisms for

- allocating roles and responsibilities for analyzing the results that an IDS produces and acting on those results
- actions that need to be taken when an alert occurs
- identifying conditions under which automated response is permitted and how the outcome of such a response is monitored to ensure the appropriate action was taken
- establishing initial anomaly-based profiles
- establishing signature-based signatures if these are not provided by the vendor or if the vendor's signatures need to be modified and expanded

Maintenance

In the maintenance phase, consider procedures and mechanisms for

- updating anomaly-based profiles to reflect current user, system, and process behavior
- updating signature-based signatures to reflect changing technology, security policies or other needs
- replacing old versions of the IDS with new versions
- maintaining awareness of ID technology improvements

Resources and Commitments

It is critical for an organization to clearly identify and assign

- all roles
- all responsibilities

- scope of authority

in order to carry out each task in each IDS life cycle phase as described above. Once these have been determined, the organization needs to identify and allocate the required staff, funding, and resources for deploying an IDS, along with the priority that these tasks have in comparison with others. Deploying and using an IDS will be successful only if the required tasks are considered as part of an organization's normal strategic and operational planning cycles and the plans for which each manager is held accountable are reviewed.

4.6 Awareness and Training

All users of an organization's information infrastructure need to become security conscious and receive periodic training if their behavior and actions are to reflect the organization's expectations with respect to security. Users include all executives, managers, staff members, business partners, vendors, contractors, and suppliers and, depending on how business is transacted, may even include customers. With training, assignment of responsibilities and authority can be made consistent with an individual's observed expertise and competence.

Security training should address the following topics:

- the organization's information security goals, objectives, policies, and procedures including sanctions for non-compliance
- secure use of information and computing resources
- how to secure the information for which users are responsible
- technical subjects such as appropriate use, password management (selection, protection, update), file access controls, expectations of privacy, user software installation, virus protection, remote access, encryption usage, and safe web browsing (specific topics should be selected from security policies and procedures)
- the nature of suspicious behavior, attacks, and intrusions (including social engineering attempts) and recognition and reporting of such events

For a complex and evolving technology such as intrusion detection, standard approaches to training (e.g., stand-up or video presentations, computer-based training, and other forms of self-paced tutorials) may not be sufficient. This is due to the need for hands-on experience in analyzing data and tracking patterns of intrusive behavior. The dynamic nature of computer technology in general, and rapidly changing threats and tactics in particular, may stress traditional forms of mentoring and on-the-job training. In other fields where immediate and intuitive response is required, simulation plays a significant role.[1] We believe that it could contribute in support of intrusion detection training [B43, B75].

1. Good analogies are learning to fly a plane or control a nuclear power plant. One must understand the theory. However, hands-on experience (as can be provided by a simulator) is essential before one is considered to be proficient.

4.7 The Decision To Make, Rent, or Buy ID Staff Capability

The term "make" indicates that the organization has hired and trained its own staff, so they are capable of selecting, deploying, operating, and maintaining an IDS. "Rent" is synonymous with "insourcing," which involves using another part of the organization (other than the one deploying the IDS) to provide the necessary staff expertise and resources, or bringing in consultants who act as part of the staff. "Buy," also known as "outsourcing," involves contracting with an organization different from the one deploying the IDS to provide all necessary ID skills and resources. In this last option, the service is usually provided remotely.

The information that ID systems produce can be extremely sensitive, which makes a case for building an organizational capability or insourcing from another business unit. However, it is increasingly difficult to attract and retain all of the necessary staff with the requisite skills to cover the full ID life cycle, particularly in light of the current market competition for people with this experience. In a CSI paper [B15], ID security experts, including Christopher Klaus of ISS and Marcus Ranum of NFR, see a growing trend towards outsourcing and make the following points:

- *Regardless of how sensitive the data is, [organizations] just don't know security very well. It all depends on how critical the data is and how security savvy the end user is. Most companies would do it themselves if they could. [Klaus]*
- *My feeling is that most users won't want to deal with these issues [how much to record, how long to keep it, and how to present it to the end user], which are complex and expensive. They'd rather buy an IDS as part of a complete network access/security package, managed by an outside agency with a 24x7 operations center. [Ranum]*
- *[Analyzing and acting upon an alert] takes a dedicated, experienced staff that sees these intrusions on a regular basis, knows how they work and more importantly, knows how to deal with them. The training and staffing requirements for this are just immense. Most companies don't have the capability, can't afford to build it, don't have time to build it, and even if they could build it, can't find the resources to build it with. [Curry]*
- *The "security skills gap" has left most organizations with little ability to really understand security at this level of technical complexity. Therefore, I think you're going to see more organizations turn to outsourcing for network security. We often hear an initial position by a client that the corporation "will not outsource security." We understand the logic and seldom raise the outsourcing issue. However, after they've seen the training requirements and the costs for 24 hour operation, they reconsider. One question often brings new light to the issue. Which would you trust more with your corporate network security? An employee who could be working for your competition next week or a service provider that is contractually bound to protect your corporate interests?*

This question usually leads to some interesting discussion. People outsource the monitoring of their home alarms. In fact, consumers have realized that home security alarm systems are of little value unless remote monitoring is involved. [Sutterfield]

4.8 Managing Expectations

Chief information officers and information system managers need to clearly set the organization's expectations regarding what the ID systems can and cannot do, particularly in light of the gaps identified in Section 3. The selection and deployment of an IDS need to be performed in the context of an overarching security architecture that reflects a layered approach to protecting an organization's assets as described earlier in this section. As a result, managers and staff should understand that an IDS has a role in protecting an organization's critical information assets but is only part of the information security solution, not a silver bullet. In fact, the topics in this section describe much higher priority security measures with greater cost/benefit advantages that should be adopted before considering the use of an IDS.

5 What Are Some Recommended Next Steps?

This section contains twenty-two recommendations for various ID communities, including potential research sponsors, users, vendors, and researchers.

5.1 Recommendations for Research Sponsors

Recommendation 1: As a result of this study, we found no operational environments in which commercial ID systems have been comprehensively tested. Test data has been generated at Lincoln Labs [Appendix E, R92], and this has been used for offline evaluation of DARPA-funded ID systems [B38]. AFRL's real time IDS evaluation environment is also being used to evaluate some of the DARPA-funded systems [B38]. Other on-going research exists at CMU [R25]. On the commercial side, some limited testing of ID systems has been performed [S20, S21]. With the increasing emphasis on COTS systems within the DoD, we recommend that an effort be undertaken to periodically test commercial ID systems using a testbed similar to the one developed by AFRL, either by AFRL or by some other agency or contractor. We state "periodic" since ID technology is constantly changing. New products and product upgrades are being released continually. New attacker exploits are constantly surfacing, and new tools and techniques that automate attacks are made publicly available. This constant state of flux means that the testing environment must be continuously upgraded and the evaluations repeated frequently so that results will be current. Current work sponsored by AFRL forms an excellent basis for an ongoing evaluation. While this evaluation task would be based on experience with DARPA-funded systems, it would be operationally oriented, not research oriented, once it is in place and operating.

We recommend that AFRL, DARPA, or another appropriate agency establish (inhouse or via contract) an independent, third-party organization to perform periodic quantitative tests on new products or major upgrades of existing products, and to account for new attack scenarios and vulnerabilities. All aspects of the evaluations (test environment, test data, scenarios, results) should be made available to the IDS community, both research and commercial. Similar evaluations should be made on GOTS systems.

To the maximum extent possible, the evaluations should be identical to those made on COTS systems and the results shared with the commercial and research communities.

Such an undertaking will require ongoing funding and some consideration should be given to equitable ways of obtaining the necessary funds from those who benefit from the activity so that the facility does not become a permanent drain on research funding.

Recommendation 2: A second crucial issue is that of high false alarm rates. The simple approaches that most of today's commercial ID systems use to detect attacks are, in most cases, unreliable. Even a very low frequency of false alarms can obscure true attack signals. Improvements in diagnostic accuracy are critically needed. We suspect that this can only be accomplished through fundamental research designed to develop a better understanding of attacks and their manifestations. ID systems research has most recently focused on architectural issues, autonomous response, and other "advanced" issues at the expense of addressing how to more accurately detect and diagnose attacks. We believe research funding organizations need to re-emphasize fundamental approaches to reduce false alarms, and recommend that research in this area should be given priority. There is a strong connection between this research area and that of testing; we believe this relationship should be encouraged.

5.2 Recommendations for Users

Users are organizations that deploy ID systems to protect their operations and computing infrastructures. Recommendations are presented for managers and purchasers to aid in selection and decision making, and for administrators and operators to aid in the operational use of ID systems.

5.2.1 Decision Makers and Buyers

Decide whether you should invest in an intrusion detection system. ID technology can help with security but it is not a panacea, and it cannot overcome inadequate system management. Unlike virus detection systems, ID systems can be complex to install and maintain. The high false alarm rates of today's systems require that significant human resources be allocated and trained to assess the validity of the results. If an organization has limited resources, these resources are better spent configuring systems securely and making sure that security policies are rigorously enforced.

Recommendation 3: Review the organizational issues described in Sections 4.3, 4.4, 4.6, and 4.7 as the first priority, before deploying a specific ID technology. Topics covered in these sections include management sponsorship, security policy, awareness and training, the make versus buy decision for ID-experienced staff, and setting appropriate management expectations.

Recommendation 4: Implement an ID strategy that reflects a defense-in-depth or layered approach to protecting an organization's assets. This includes defining and enforcing an effective security policy, installing one or more firewalls for limiting access, and deploying access control and user authentication mechanisms. Further details on the elements of a layered approach can be found in Section 4.5.

Recommendation 5: Select an ID system that meets your needs. Address the following topics to aid your selection process:

- Develop ID requirements based on security policy and organizational needs. An organization cannot define what constitutes an intrusion (and therefore what functionality the ID system should provide) without the context set by security policy. ID requirements should address the organization's highest risk threats, such as detecting intrusions resulting from internal threats, external threats with a single point of entry, or external threats with multiple and VPN points of entry.
- Assess the need for both network- and host-based solutions for comprehensive coverage. Host and network-based solutions have their respective strengths and weaknesses and the strengths of one approach tend to complement the weaknesses of the other.

Use network-based ID systems to

- detect known attacks and take appropriate automated response actions,
- monitor firewall policy by placing an ID system on each side of the firewall,
- detect denial of service attacks, monitor specific services, servers, and protocols

Use host-based ID systems to

- determine normal behavior and detect unexpected behavior (departures from normal beyond some threshold) at the level of systems, users, processes, and applications
- deal with encrypted traffic
- focus on critical assets and their protection

- Regularly review the trade and testing literature on commercial ID systems. Be aware that trade literature, particularly in this field, tends to be full of marketing hype. Critical information on, for example, exploit signature characteristics and their accuracy is generally not provided. Independent tests of ID systems are available [S20, S21, S24, S25] and can be useful. However, few reviews provide quantitative results on false alarm rates and they are quickly out-of-date. It is not advisable to rely on reviews that are more than three months old unless you verify the correctness of the information. Refer to Section 4.4, Appendix F, and "Tough Questions for IDS Vendors" [B14] for more information.

5.2.2 Administrators and Operators

Recommendation 6: Configure the ID system to maximize performance. Include the following:

- Selectively deploy host-based ID systems in close proximity to critical assets (e.g., DNS and application servers). Deploying these systems on all hosts may be impractical, cost prohibitive, and can adversely affect performance. Selective deployment allows administrators to focus their limited resources.
- Identify and install only those signatures that correlate strongly with unique attack scenarios (i.e., where the probability of false positives is to close to zero). Configure the ID system to take automated response actions based on what it detects (due to having high confidence in this class of alerts).
- Tune the network ID system to prevent excessive false alarms and to identify events of most interest.
- Explicitly suppress all classes of attacks, scans, etc. that are made irrelevant through the elimination of unnecessary services.
- Specify only attack signatures or profiles that reflect vulnerabilities relevant to the networks and systems to be protected.[1]

Recommendation 7: Collect and analyze data, and respond to intrusions using well defined and documented procedures [B98, B123] that address

- analyzing ID results based on trends in alert message traffic and their relationship to some predefined thresholds. There is usually low payoff in analyzing individual messages unless statistical trends are first reviewed.
- maintaining system logs
- gathering forensic evidence
- responding to intrusive activity
- backing up data
- recovering from an intrusion
- interfacing with external organizations (other system administrators, law enforcement agencies such as the FBI, security centers such as CERT/CC and CIAC)

1. We recognize that configuring out certain operations may not always be realistic given the complexity of today's products.

5.3 Recommendations for Vendors

The vendor community appears to be focused on unnecessarily narrow approaches to intrusion detection. We believe this is due, in part, to customers who are motivated more by their management's desire to state, "we have an ID system in place; therefore, we have done the best we can," rather than to seriously look at the effectiveness of such ID systems. This attitude appears to prioritize concerns for liability over security.

One engineer from a well-known vendor organization stated that he cannot implement any new features unless such features are mandated by customer demand as reflected by the vendor marketing department. This problem is exacerbated by the highly competitive environment forcing vendors to focus their new development efforts on meeting short-term needs.

If the ID industry is to mature and prosper, a broader perspective must be adopted. One technology that has succeeded is virus-detection, and there are lessons to be learned from this field. Several reasons why an anti-virus company could excel in the ID industry include [B76]

- No security tool has better computer desktop penetration than anti-virus software.
- ID tools have 200 or fewer signatures; anti-virus software can detect more than 20,000.
- Anti-virus software comes with implementations for firewalls, server systems, or computer desktops.
- Anti-virus software can identify, contain, eradicate, and recover with minimal user intervention.
- Anti-virus companies have fully solved the issue of updating a user's signature table with a variety of painless options.
- Many large organizations have site licenses with these software companies and are satisfied.
- Anti-virus companies are already oriented to fast turnaround of a signature table when a new exploit is detected.
- These software companies often have companion products with security capabilities.

Recommendation 8: Vendors need to support initiatives to create open source signatures and to move the ID community towards the distribution model used by the anti-virus community. This action would allow

- users of signatures to understand exactly what is being detected and the effectiveness of the signature
- public discourse, likely resulting in more robust signatures
- the creation of public signature repositories to the community's benefit

We recognize that such disclosure is unlikely to happen in the current competitive environment. However, tools such as NFR [C2] and Snort™ [C24] do operate with an open signature source policy; we hope that these examples will influence others to follow.

Recommendation 9: Spend more time and resources testing signatures. The current practice of keeping signatures proprietary encourages ineffective testing. Black box testing (the only approach to testing when access to actual signatures is denied) cannot provide the insights gained through white box testing. Unfortunately, it appears that the drive to release new signatures is greater than the drive to assure signature accuracy.

Recommendation 10: Provide measures that represent the level of confidence a user should place in an ID system's ability to report an intrusion based on matching a signature — by type of signature or type of attack. Some signatures detect intrusions with almost perfect accuracy (i.e., there are virtually no false positives), some are absolutely questionable (such as SYN flood or slow port scan), and some require greater or lesser degrees of manual analysis in addition to what the ID system reports (such as packet fragmentation). A higher level of confidence in the diagnosis (and many fewer false alarms) would significantly extend the usefulness of an ID system.

Recommendation 11: Partly because of customer pressure, vendors attempt to fully automate intrusion diagnosis. A more realistic approach is to involve the human in the diagnostic loop. While computers are capable of examining large quantities of low level data, they cannot match a human's analytic skills. Such a system might, at a minimum, provide graphical displays of statistical patterns for human interpretation.

Recommendation 12: Integrate data sources more effectively. Current commercial ID systems perform relatively simple pattern matching as a basis for detecting intrusions. However, with the increasing sophistication of attacks (e.g., distributed coordinated attacks), such an approach will provide diminishing returns.

Approaches to consider include the following:

- Most stand-alone network-based systems or host-based systems do not integrate data from different sensors. In particular, we believe that application-based ID monitoring should be built into applications and that a host-based ID system should then correlate events across applications logs.
- Many vendors are scrambling to distribute products that have both network- and host-based components. However, in the rush to release this combination, not much thought is given to integration. The outcome is that network and host alerts are presented to the security administrator with little thought given to integrated diagnosis. This not only overloads the administrator, but does not fully utilize all sensor data in making the most accurate diagnosis.

- The presence of specialized, lightweight sensors tailored to detect specific intrusions or intrusions against specific applications may be a way to improve detection accuracy if the sensor results are properly integrated into the ID system
- More emphasis needs to placed on the integration of information from different ID systems. This includes a need for systems from different vendors to interoperate at all levels. There is emerging work in this area.

Recommendation 13: Consider expanding options for capturing forensic evidence. ID systems have the potential to filter large quantities of data and provide information that can more directly support forensic analysis.
If a network ID system identifies a signature suggesting an intrusion, it could selectively record the relevant packets for subsequent legal examination. ID systems could support time-lining of suspicious activities such as network scans, login attempts, and document modification. Refer to Section 3.5.3 for more details.

Recommendation 14: Concern over the security aspects of mobile code is increasing. While detection of malicious mobile code (email attachments, Java, ActiveX) has not been the traditional focus of ID systems, we recommend it for vendor consideration. This topic is further discussed in Section 3.1.7.

Recommendation 15: Increase interaction with the research community. Techniques being developed by the research community may provide useful guidance for future commercial products. Some of these directions are identified in Sections 3 and 5.4.

In summary, the future of ID technology is not in new, broad-spectrum sensor development but in more rigorous testing of signatures, reducing false alarm rates, developing lightweight sensors targeted at specific threats, providing improved management and user interface capabilities, and effectively integrating the results of network- and host-based ID sensors within a single vendor tool suite and among tools from different vendors.

5.4 Recommendations for Researchers

Recommendation 16: Perhaps the most serious criticism leveled at current ID technology is the high false alarm rate. More sophisticated analysis needs to be performed, including integrating diverse sources of available data. For sophisticated attacks, information from one sensor is unlikely to detect suspicious activity. This issue is more fully described in Section 3.6.4 and Recommendation 2, Section 5.1. Some of the vendor recommendations in Section 5.3 may be applicable to the research community for addressing this issue.

Recommendation 17: ID systems are not adequately tested. Several efforts have focused on developing test beds and generating test data. However, most have addressed non-commercial tools. The ID research community needs to develop environments in which it is possible to

- rapidly generate test data reflecting new attacks and vulnerabilities
- rapidly evaluate new or upgraded commercial systems against that data.

Unfortunately, the vendor community lacks the motivation to support public evaluation of its tools. Additional discussion of this issue can be found in Section 3.5.1 and is also recommended for research sponsor consideration in Section 5.1.

Recommendation 18: Many taxonomies in the ID field have described types of attack from the attacker perspective. For building ID systems, a taxonomy of vulnerabilities would be more useful, as this provides greater insights in how to construct effective defenses. By focusing on vulnerabilities, there is a greater likelihood that classes of attacks can be defended against as opposed to single attacks. We believe that further efforts to clarify the relationship between vulnerabilities will provide insights that make ID systems more robust. The issue of taxonomies is discussed further in Section 3.3.4.

Recommendation 19: Increasingly, sophisticated attackers will target ID systems themselves. Awareness of this issue with respect to network-based ID systems was raised in the paper by Ptacek and Neusham [B26-b] and discussed in more detail in Section 3.1.5. We recommend that detecting or defending against such attacks (insertion, evasion and denial of service) become a high priority research issue.

Recommendation 20: The need for human involvement in supporting intrusion detection was discussed with respect to vendors. However, researchers also need to address the following questions:

- In what ways can past human experience in detecting intrusions be more effectively incorporated in to ID systems?
- Regarding intrusion detection, can human mental models of analysis shed light on human-computer interaction?
- What types of graphical displays or other or other devices are most effective in supporting human analysis of ID data?

These issues are more fully explored in Section 3.3.2.

Recommendation 21: The popularity of mobile code is growing rapidly, introducing significant vulnerability. Such code can bypass many security measures. Evaluating mobile code in detail as seen by the host may be the only means by which to determine if it is malicious. Much effort has gone into controlling malicious behavior of Java code, but this has not been completely successful. Other means of delivering mobile code (through email attachments or ActiveX) are even less secure. As indicated in Section 3.1.7, we believe that there are several ways in which ID systems can help.

Recommendation 22: The research and the vendor community need to more closely cooperate. Any organized effort in this area should probably come from funding agencies and program managers (though contacts by individual researchers should also be encouraged). Information sharing would be helped by informing representatives of commercial companies about ID related workshops, symposia, program reviews, and by inviting them to these and similar events. Such invitations should be issued on a regular basis, whether a substantial number of vendors choose to participate or not.

Appendix A: Glossary

All of these terms are defined within the context of information assurance.

adversary

From a defender's viewpoint: one who is expected to attack an asset for which you are responsible. A group or organization. In military, national security, a nation state; a terrorist group. In autonomous agents, a software agent also known as a malicious agent. From an attacker's viewpoint: one who is responsible for defending the asset you intend to attack. Therefore, attackers and defenders are mutual adversaries. See also: attacker, intruder, victim.

analysis approach

A method used by an IDS to determine whether or not an intrusion has occurred. The two approaches defined below (attack signature detection, anomaly detection) are based on what the deployer or operator of the IDS knows prior to installing the IDS. In the following descriptions, "you" refers to the deployer/ operator:

- You know what bad things (i.e., known attacks) you want to detect. Knowing this, you use attack signature detection which may identify that a bad thing has happened or may require further analysis (manual inspection) to make the determination
- You know what constitutes acceptable user, process, or system behavior. You want to know when behavior occurs that is outside the bounds of some acceptable threshold or measure. Such behavior constitutes an abnormal occurrence (i.e., an event of interest), not necessarily an attack.

attack signature detection

Identifies patterns corresponding to known attacks. This includes both passive protocol analysis (use of sniffers in promiscuous mode) as well as signature analysis (the interpretation of a certain series of packets, or a certain piece of data contained in those packets, that are determined, in advance, to represent a known pattern of attack) [B26-b].

anomaly detection

Identifies any unacceptable deviation from expected behavior. Expected behavior is defined, in advance, by a manually-developed profile or by an automatically-developed profile. An automatically-developed profile is created by software that collects and processes characteristics of system behavior over time and forms a statistically valid sample of such behavior. Some of these deviations do not require further examination and some do. An anomaly might include

- users logging in at strange hours
- unexplained reboots or changes to system clocks
- unusual error messages from mailers, daemons, or other servers
- multiple, failed login attempts with bad passwords
- unauthorized use of the su command to gain UNIX root access
- users logging in from unfamiliar sites on the network

One approach to anomaly detection is statistical analysis. A subcategory of anomaly detection is integrity checking which determines whether some aspect of a file or object has been altered.

Anomaly detection assumes that intrusions are highly correlated to abnormal behavior exhibited by either a user or an application. The basic idea is to baseline normal behavior of the object being monitored and then flag behaviors that are significantly different from the baseline as abnormalities, or possible intrusions. [R61]

attack (noun)

An action conducted by an adversary, the attacker, on a potential victim. A set of events which an observer believes to have information assurance consequences on some entity, the target of the attack. From the perspective of an administrator responsible for maintaining a system, an attack is a set of one or more events that has one or more security consequences. From the perspective of a neutral observer, the attack can either be successful, an intrusion, or unsuccessful, an attempted or failed intrusion. From the perspective of an intruder, an attack is a mechanism to fulfill an objective. It is unclear whether an unsuccessful attack is an intrusion. Intrusion seems to imply forced entry, while attack seems to only imply the application of force. Host xyz attacked site 123 with the blatsit attack. See also: intrusion, adversary, intruder, target, victim.

attack (verb)

To begin to act upon destructively, to begin to destroy, expose, alter, or disable.

attacker

An adversary who conducts an attack on a victim (e.g., host). Contrast with intrude. See also: intruder, attack.

auditing

Systematically examining system data against documented expectations of form or behavior to verify conformance with documented expectations.

availability

The property of a system or a system resource being accessible and usable upon demand by an authorized system entity, according to performance specifications for the system; i.e., a system is available if it provides services according to the system design whenever users request them.

confidentiality

The property that information is not made available or disclosed to unauthorized individuals, entities, or processes (i.e., to any unauthorized system entity)

consequence

The change in security caused by a particular action. In vulnerability analysis, the theoretical change in system state (i.e., the state transition itself) that would be effected on a system of a particular class assuming that the system has the vulnerability being analyzed and that the vulnerability were exploited. In artifact analysis, the theoretical change in system state (i.e., the state transition itself) that would be effected on a system of a

particular class assuming that the particular artifact (i.e., attack program) being analyzed was executed successfully against the system. The consequence of an action may be quite different depending on the observer's frame of reference (e.g., attacker, victim, uninvolved). Contrast consequence with impact. For a given action, its consequence is the terminal state and its impact is the state transition from the starting system state to the terminal system state.

Consequence includes unauthorized access, unauthorized use, denial of service, reconnaissance deception, alteration of data, destruction of data.

exploit (verb)

To, in some way, take advantage of a vulnerability in a system in the pursuit or achievement of some objective. All vulnerability exploitations are attacks but not all attacks exploit vulnerabilities.

exploit (noun)

Colloquially for exploit script: a script, program, mechanism, or other technique by which a vulnerability is used in the pursuit or achievement of some information assurance objective. It is common speech in this field to use the terms exploit and exploit script to refer to any mechanism, not just scripts, that uses a vulnerability.

false negative

Occurs when the IDS fails to identify an intrusion when one has in fact occurred [B26-b].

false positive

Occurs when the IDS incorrectly identifies an intrusion when none has occurred [B26-b].

impact

The negative effect of an attack on a victim system by an attacker. In incident analysis, the negative effect on a system that results from exploiting a particular vulnerability; as in the vulnerability's impact. In vulnerability analysis, the hypothesized negative state that would be effected on a system of a particular class assuming that the system has the vulnerability in question and that the vulnerability were exploited. The use of this term occurs most frequently in incident analysis. In vulnerability analysis, its use is generally deprecated in favor of the more precise consequence. See also: consequence, attack, vulnerability.

incident

A collection of data representing one or more related attacks. Attacks may be related by attacker, type of attack, objectives, sites, or timing.

information assurance

The subfield of information science that focuses on the conditions necessary to assure users of information systems and services that they can expect:

1. the information and services they use actually did originate with whom they claim and are exactly as the originator intended
2. the information and services they use will be available when needed
3. the information and services for which they are responsible will be made available only to those they intend and only in the manner that they intend

See also: security, survivability.

integrity

For systems, the quality that a system has when it can perform its intended function in a unimpaired manner, free from deliberate or inadvertent unauthorized manipulation.

For data, the property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner.

inspection

Examining a data resource or process to identify anomalous content or behavior in the data resource or process.

intruder

An adversary who is conducting or has conducted an intrusion or attack against a victim host, site, network, or organization. Since the label of intruder is assigned by the victim of the intrusion and is therefore contingent on the victim's definition of encroachment, there can be no ubiquitous categorization of actions as being intrusive or not. From the victim's viewpoint, an intruder is usually an entity (person or organization) that has successfully attacked the victim. It is unclear whether one who conducts an unsuccessful attack is an intruder. If an intrusion is required to be an intruder, then it seems that all intruders are attackers, but all attackers are not necessarily intruders.

See also: attacker, adversary, attack, intrusion.

intrusion

Actual illegal or undesired logical entry into an information system; The act of violating the security policy or legal protections that pertain to an information system.

The concept of an unsuccessful intrusion is not common in colloquial English, but it is common in information assurance. This causes ambiguity in use in information assurance. Some uses seem to encompass both successful and unsuccessful intrusions, while others seem to imply successful intrusions only and unsuccessful intrusions are just attacks. It is unclear whether an unsuccessful attack is an intrusion. Intrusion seems to imply forced entry, while attack seems to only imply the application of force. See also: attack.

intrusion detection system

A combination of hardware and software that monitors and collects system and network information and analyzes it to determine if an attack or an intrusion has occurred. Some ID systems can automatically respond to an intrusion.

intrusion detection technologies

A broader term (than intrusion detection system) meaning a combination of ID systems, intrusion analysts, and other supporting tools (such as those that process raw network packets or log files). Used together, ID technologies can provide accurate indicators of whether or not an attack or intrusion has occurred.

logging

Systematically recording specified events in the order that they occur to provide a data trail for subsequent analysis.

mission

A set of very high-level (i.e., abstract) requirements or goals. Missions are not limited to military settings since any successful organization or project must have a vision of its objectives whether expressed implicitly or as a formal mission statement. Judgments as to whether or not a mission has been successfully fulfilled are typically made in the context of external conditions that may affect the achievement of that mission.

monitoring

Observing a data stream for specified events to provide data for subsequent action or analysis.

response Actions taken to protect and restore the normal operating condition of computers and the information stored in them when an attack or intrusion occurs. Also referred to as incident response or intrusion response.

security The subfield of information science concerned with ensuring that information systems are imbued with the condition of being secure, as well as the means of establishing, testing, auditing, and otherwise maintaining that condition.

site A logical group of interconnected physical machines all under the control of a single administrative unit. The administrative unit itself. The DNS domain name of the administrative unit. A site almost always contains multiple systems. Any one of these systems is not necessarily wholly contained within the site. Generally, the machines that make up a site are collocated geographically, hence the name. A site is frequently equated with its DNS domain name. James attacked site blue. James attacked blue.com.

survivability The capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents. Timeliness is a critical factor that is typically included in (or implied by) the very high-level requirements that define a mission. However, timeliness is such an important factor that we include it explicitly in the definition of survivability. It is important to recognize that it is the mission fulfillment that must survive, not any particular subsystem or system component. Central to the notion of survivability is the capability of a system to fulfill its mission, even if significant portions of the system are damaged or destroyed. We will sometimes use the term survivable system as a less than perfectly precise shorthand for a system with the capability to fulfill a specified mission in the face of attacks, failures, or accidents. See also: attack, mission, system.

system one or more interconnected physical machines (hosts) operating in cooperation with one another to meet a particular mission. Systems are generally, although not necessarily, contained within one site. Hosts may participate in multiple systems. Systems may be wholly contained within one host or distributed across multiple hosts. See also: mission.

target (noun)

The object of an attack, especially host, computer, network, system, site, person, organization, nation, company, government, or other group. See also: attack, victim.

target (verb)

To use something or someone as a target. To plan or schedule something or someone to attain an objective. For many computer-based attacks, target selection and attack are tightly integrated and, perhaps, indistinguishable. See also: attack.

victim

That which is the target of an attack. An entity may be a victim of either a successful or unsuccessful attack. See also: adversary, attack, attacker, intruder, intrusion.

vulnerability

A feature or a combination of features of a system that allows an adversary to place the system in a state that is both contrary to the desires of the people responsible for the system and increases the risk (probability or consequence) of undesirable behavior in or of the system. A feature or a combination of features of a system that prevents the successful implementation of a particular security policy for that system. A program with a buffer that can be overflowed with data supplied by the invoker will usually be considered a vulnerability. A telephone procedure that provides private information about the caller without prior authentication will usually be considered to have a vulnerability.

Appendix B: Bibliography

Please note that some references have suffixes (e.g., B25-1 or R23-b). If the suffix is a number, you can find all of the references in the series on the same Web site. If the suffix is a letter, the sources are associated with each other but are found in different locations. If a reference such as R25 is cited and R25-1, R25-2, etc. exist in the bibliography, then the citation refers to the entire group.

In the course of preparing this report, some bibliographic references were deleted or combined with others. Please note that in some cases this created a gap in the numeric sequence found in the references (e.g., B19, B20, B23).

This report contains many Web references. The intrusion detection field changes rapidly and much information is posted first (and often only) on the Web. Many of these references either become out of date, are modified, or disappear altogether from the original site. If you have questions or comments about information in this report, please send email to security-improvement@cert.org.

GENERAL INFORMATION

[B1] DARPA. *Intrusion Detection PI Meeting December 1998—Agenda* [online]. Available WWW: <URL: http://www.dyncorp-is.com/darpa/meetings/id98dec/agenda.html> (1998).

[B2] DARPA. *Intrusion Detection PI Meeting February 1998—Agenda and Presentations* [online]. Available WWW: <URL: http://www.dyncorp-is.com/darpa/meetings/id98feb/agenda.html> (1998).

[B3] Sobirey, Michael. *Michael Sobirey's ID Systems Page* [online]. Available WWW: <URL: http://www-rnks.informatik.tu-cottbus.de/~sobirey/ids.html> (1999).

[B4] Stocksdale, Gregory. (National Security Agency). *SANS/NSA Intrusion Detection Tools Inventory* [online]. Available WWW: <URL: http://www.sans.org/NSA/idtools.htm> (1998).

[B5] DARPA & Air Intelligence Agency. *CSAP21—Information Protection into the 21st Century* [online]. Available WWW: <URL: http://www.darpa.mil/iso/ia/ssd/FutureTech/sld001.htm> (1998).

[B6] Stocksdale, Gregory. (National Security Agency). *NSA Glossary of Terms in Security and Intrusion Detection* [online]. Available WWW: <URL: http://www.sans.org/NSA/glossary.htm> (1999).

[B7] Puketza, Nicholas, et al. "A Software Platform for Testing Intrusion Detection Systems." *IEEE Software 14*, 5: 43-51 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/papers.html> (1997).

[B8] Heberlein, L. Todd & Bishop, Matt. "Attack Class: Address Spoofing." *Proceedings of The 19th National Information Security Conference* [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/papers.html> (1996).

[B9] Puketza, Nicholas J., et al. (University of California, Davis). "A Methodology for Testing Intrusion Detection Systems." *IEEE Transactions on Software Engineering, Vol. 22, #10 (SE-22)* (October 1996): 719-729 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/papers.html>.

[B10] Kahn, Clifford, et al. *A Common ID Framework* [online]. Available WWW: <URL: http://www2.csl.sri.com/intrusion> (1998).

[B11] Network Associates Security Labs. *Evading Intrusion Detection—Executive Summary* [online]. Available WWW: <URL: http://www.nai.com/products/security/advisory/papers/ids-simple.doc> (1999).

[B12] Peter Davis & Associates. *Intrusion Detection Systems* [online]. Available WWW: <URL: http://www.pdaconsulting.com/ids.htm> (1997).

[B13] Hurwitz Group, Inc. *Information Security: Assessing Risks and Detecting Intrusions* [online]. Available WWW: <URL: http://www.summitonline.com/security/papers/hurwitz3.html> (1998).

[B14] Computer Security Institute. *Tough Questions for IDS Vendors* [online]. Available WWW: <URL: http://www.gocsi.com/IDSques.htm> (1998).

[B15] Power, Richard. "CSI Round Table: Experts Discuss Present and Future Directions for ID Systems." *Computer Security Journal XIV*, 1 [online]. Available WWW: <URL: http://www.gocsi.com/roundtable.htm> (1999).

[B16] Debar, H., et al. (IBM Zurich). *An Experimentation Workbench for Intrusion Detection Systems* (RZ2998). Zurich, Switzerland: IBM Research Division, March 1998 [online]. Available WWW: <URL: http://www.zurich.ibm.com/pub/sti/Security/extern/gsal/docs/>.

[B18] Taber, Mark. "The Sams Crack Level Index," Ch. 26 "Levels of Attack." *Maximum Security: A Hacker's Guide to Protecting Your Internet Site and Network* [online]. Available WWW: <URL: http://www.damocles.com/~kronvold/Hacker/docs/v0000027.htm> (1999).

[B19] Van Doorn, Leendert. (Vrije Universiteit, Amsterdam). *Computer Break-ins: A Case Study* [online]. Available WWW: <URL: http://www.alw.nih.gov/Security/FIRST/papers/general/holland.ps> (1999).

[B20] Chung, M., et al. (University of California, Davis). "Simulating Concurrent Intrusions for Testing Intrusion Detection Systems: Parallelizing Intrusions," 173-183. *Proceedings of the 1995 National Information Systems Security Conference*. Baltimore, MD, October 10-13, 1995. [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/papers.html>.

[B23] Bace, Rebecca. (Infidel, Inc.). *An Introduction to Intrusion Detection and Assessment* [online]. Available WWW: <URL: http://solutions.iss.net/products/whitepapers/intrusion.pdf> (1999).

[B24] Internet Security Systems, Inc. *Real-Time Attack Recognition and Response: A Solution for Tightening Network Security* [online]. Available WWW: <URL: http://solutions.iss.net/products/whitepapers/realtime.pdf> (1999).

[B25] Internet Security Systems, Inc. *Network- vs. Host-based Intrusion Detection* [online]. Available WWW: <URL: http://solutions.iss.net/products/whitepapers/nvh_ids.pdf> (1998).

[B26-a] Cohen, Fred. *50 Ways to Defeat Your Intrusion Detection System* [online]. Available WWW: <URL: http://www.all.net/journal/netsec/9712.html> (1999).

[B26-b] Ptacek, Thomas H. & Newsham, Timothy N. (Secure Networks, Inc.) *Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection* [online]. Available WWW: <URL: http://www.clark.net/pub/roesch/public_html/IDSpaper.pdf> (1998).

[B27] Cohen, Fred. *Anatomy of a Successful Sophisticated Attack* [online]. Available WWW: <URL: http://www.all.net/journal/netsec/9901.html> (1999).

[B28] Cheswick, Bill. (AT&T Bell Laboratories). *An Evening With Berford in Which a Cracker is Lured, Endured and Studied* [online]. Available FTP: <URL: http://jhunix.hcf.jhu.edu/pub/miscellaneous_security_papers/An_Evening_With_Berferd.ps.Z> (1999).

[B29] MIT Lincoln Laboratory. *DARPA Intrusion Detection Evaluation* [online]. Available WWW: <URL: http://www.ll.mit.edu/IST/ideval/index.html> (1999).

[B30] Zissman, Marc A. & Lippmann, Richard P. (MIT Lincoln Laboratory). "Intrusion Detection System Evaluation." *IA Newsletter 2,2* (Fall 1998): 6-7.

[B31] Tobin, Donald L., Jr. (University of Idaho). "Detecting Intrusions Cooperatively Across Multiple Domains." *IA Newsletter 2,2* (Fall 1998): 10.

[B33-1] Levitt, Karl. (University of California, Davis). "Executive Summary." *Proceedings of the 4th Workshop on Future Direction in Computer Misuse and Anomaly Detection (CMAD IV).* Monterey, CA, Nov. 12-14, 1996 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/cmad/4-1996/exec_summ.html>.

[B33-2] Sharps, Jennifer. (University of California, Davis). "Session 1: Policy-Driven Intrusion Detection and the Insider Threat." *Proceedings of the 4th Workshop on Future Direction in Computer Misuse and Anomaly Detection (CMAD IV).* Monterey, CA, Nov. 12-14, 1996 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/cmad/4-1996/session1.html>.

[B33-3] Levitt, Karl. (University of California, Davis). "Session 2: Intrusion Detection Technology for Small-Scale Systems." *Proceedings of the 4th Workshop on Future Direction in Computer Misuse and Anomaly Detection (CMAD IV).* Monterey, CA, Nov. 12-14, 1996 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/cmad/4-1996/session2.html>.

[B33-4] Wee, Christopher & Heberlein, Todd. (University of California, Davis). "Session 3: New Attacks and New Twists on Existing Attacks." *Proceedings of the 4th Workshop on Future Direction in Computer Misuse and Anomaly Detection (CMAD IV).* Monterey, CA, Nov. 12-14, 1996 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/cmad/4-1996/session3.html>.

[B33-5] Spafford, Gene. (University of California, Davis). "Session 4: Intrusion Detection in the Large." *Proceedings of the 4th Workshop on Future Direction in Computer Misuse and Anomaly Detection (CMAD IV).* Monterey, CA, Nov. 12-14, 1996 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/cmad/4-1996/session4.html>.

[B33-6] Schaefer, Marv & Levitt, Karl. (University of California, Davis). "Session 5: New Environments for Intrusion Detection." *Proceedings of the 4th Workshop on Future Direction in*

Computer Misuse and Anomaly Detection (CMAD IV). Monterey, CA, Nov. 12-14, 1996 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/cmad/4-1996/session5.html>.

[B33-7] Bace, Becky. (University of California, Davis). "Session 6: Tools for Investigative Support." *Proceedings of the 4th Workshop on Future Direction in Computer Misuse and Anomaly Detection (CMAD IV).* Monterey, CA, Nov. 12-14, 1996 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/cmad/4-1996/session6.html>.

[B33-8] Gragg, Susan. (University of California, Davis). "Session 7: New Ideas." *Proceedings of the 4th Workshop on Future Direction in Computer Misuse and Anomaly Detection (CMAD IV).* Monterey, CA, Nov. 12-14, 1996 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/cmad/4-1996/session7.html>.

[B34] Anderson, James P. *Computer Security Threat Monitoring and Surveillance.* Fort Washington, PA: James P. Anderson Co., 1980.

[B35] Horizon. "Defeating Sniffers and Intrusion Detection Systems." *Phrack Magazine 8*, 54 (Dec. 25, 1998): article 10 of 12 [online]. Available WWW: <URL: http://pulhas.org/phrack/54/P54-10.html>.

[B37] Ranum, Marcus J. *Security on Internet Time* [online]. Available WWW: <URL: http://www.clark.net/pub/mjr/pubs/index.shtml> (1997).

[B38] Durst, Robert, et al. "Testing and Evaluating Computer Intrusion Detection Systems." *Communications of the ACM 42*, 7 (July 1999): 53-61.

[B41] Mansur, Doug L. (Lawrence Livermore National Laboratory). *Current Trends in the Threat to Computers: From Simple Hacking to Cyber Terrorism* [online]. Available WWW: <URL: http://doe-is.llnl.gov/SecRes/DOETools/99001latalk.pdf> (1998).

[B42-1] Erlinger, Michael, et al. *Intrusion Detection Exchange Format (idwg)* [online]. Available WWW: <URL: http://www.ietf.org/html.charters/idwg-charter.html> (1999).

[B42-2] Erlinger, Mike & Staniford-Chen, Stuart. *IDWG Charter* [online]. Available WWW: <URL: http://www.zurich.ibm.com/Technology/Security/extern/idwg/charter.html> (1998).

[B42-3] Hoffman, Paul. (Internet Mail Consortium). *A Novice's Guide to the IETF* [online]. Available WWW: <URL: http://www.imc.org/novice-ietf.html> (1999).

[B43] Cohen, Fred. *Simulating Network Security* [online]. Available WWW: <URL: http://www.all.net/journal/netsec/9904.html> (1999).

[B44] Cohen, Fred. *Returning Fire* [online]. Available WWW: <URL: http://all.net/journal/netsec/9902.html> (1999).

[B45-a] ICSA.net. *About ICSA* [online]. Available WWW: <URL: http://www.icsa.net/about_icsa/> (1999).

[B45-b] SANS Institute Online. *SANS Institute Online—Home Page* [online]. Available WWW: <URL: http://www.sans.org/newlook/home.htm> (1999).

[B45-c] Information Assurance Technology Center (IATAC). *About IATAC* [online]. Available WWW: <URL: http://www.iatac.dtic.mil/About.htm> (1999).

[B45-d] The Internet Engineering Task Force (IETF). *Overview of the IETF* [online]. Available WWW: <URL: http://www.ietf.org/overview.html> (1999).

[B45-e] Staniford-Chen, Stuart. *Common Intrusion Detection Framework (CIDF)* [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/cidf/> (1998).

[B45-f] Security Research Alliance. *Security Research Alliance—Overview* [online]. Available WWW: <URL: http://www.securityresearch.com/overviewmain.htm> (1999).

[B47] Jajodia, S.; McCollum, C.D.; & Ammann, P. "Trusted Recovery." *Communications of the ACM 42,* 7 (July 1999): 71-75.

[B48] Spafford, E.H. & Weeber, S.A. "Software Forensics: Can We Track Code to Its Authors?" 641-650. *Proceedings of the 15th National Computer Security Conference.* Oct 13-16, 1992. (Coast TR 91-01) [online]. Available WWW: <URL: http://www.cs.purdue.edu/coast/coast-library.html>.

[B49] Kerstetter, Jim. *Low-Flying Hackers Pose Growing Threat* [online]. Available WWW: <URL: http://www.zdnet.com/pcweek/stories/news/0,4153,360254,00.html> (1998).

[B50] Robbins, Judd. *An Explanation of Computer Forensics* [online]. Available WWW: <URL: http://knock-knock.com/forens01.htm> (1999).

[B51] Denning, Dorothy E. *Who's Stealing Your Information?* [online]. Available WWW: <URL: http://www.infosecuritymag.com/apr99/cover.htm> (1999).

[B52] Cohen, Fred. *Attack and Defense Strategies* [online]. Available WWW: <URL: http://all.net/journal/netsec/9907.html> (1999).

[B53] Ranum, Marcus J. "Is Network Intrusion Detection Software Being Used Correctly?" *Security Management 42*, 8 (August 1998): 124-126.

[B54] Irwin, Vicki & Northcutt, Stephen. (Naval Surface Warfare Center, Dalgren). *Shadow: Internet Threat Briefing—Stealth & Coordinated Attacks* [online]. Available WWW: <URL: http://www.nswc.navy.mil/ISSEC/CID/coordinated.ppt> (1999).

[B55-a] Naval Surface Warfare Center, Dalgren. *SHADOW Indications Technical Analysis—Coordinated Attacks and Probes* [online]. Available WWW: <URL: http://www.nswc.navy.mil/ISSEC/CID/co-ordinated_analysis.txt> (1998).

[B55-b] Northcutt, Stephen. (Naval Surface Warfare Center, Dahlgren). "Intrusion Detection: Shadow Style—Step by Step Guide." *SANS Institute Report* (November 1988).

[B57] Amoroso, Edward & Kwapniewski, Richard. (AT&T Laboratories). "A Selection Criteria for Intrusion Detection Systems." *Proceedings of the 14th Annual Computer Security*

[B58-1] Hosmer, Chet; Feldman, John; & Giordano, Joe. *Advancing Crime Scene Computer Forensic Techniques* [online]. Available WWW: <URL: http://www.wetstonetech.com/crime.htm> (1999).

[B58-2] Hosmer, Chet. *Announcing the Formation of New High Technology Software Company* [online]. Available WWW: <URL: http://www.wetstonetech.com/pr9801.htm> (1998).

[B59-a1] Fyodor. *The Art of Port Scanning* (see also [B129]) [online]. Available WWW: <URL: http://www.insecure.org/nmap/nmap_doc.html> (1997).

[B59-a2] Fyodor. *Nmap Network Security Scanner Man Page* [online]. Available WWW: <URL: http://www.insecure.org/nmap/nmap_manpage.html> (1999).

[B59-b1] Harrison, Ann. *New Generation of Scanning Tools Mask Source of Attack* [online]. Available WWW: <URL: http://www.computerworld.com/home/print.nsf/all/99031596AA> (1999).

[B59-b2] Harrison, Ann. *When Good Scanners Go Bad* [online]. Available WWW: <URL: http://www.computerworld.com/home/print.nsf/all/9903229872> (1999).

[B59-c] Beyond Security. *NMap Port Scanner* [online]. Available WWW: <URL: http://www.securiteam.com/tools/NMap_Port_scanner.html> (1999).

[B61] Fyoder. *Remote OS Detection via TCP/IP Stack Fingerprinting* [online]. Available WWW: <URL: http://128.196.109.24/nmap/nmap-fingerprinting-article.txt> (1998).

[B62-a] Privacy.net. *Privacy Analysis of Your Internet Connection—How It Works* [online]. Available WWW: <URL: http://privacy.net/analyze/analyzehow.asp> (1999).

[B62-b] Oakes, Chris. *Cracking Tools Get Smarter* [online]. Available WWW: <URL: http://www.wired.com/news/news/technology/story/18219.html> (1999).

[B63] Neikter, Carl-Fredrik. *Netbus Pro 2.01* [online]. Available WWW: <URL: http://netbus.org/features.html> (1999).

[B64-a] LaMonaca, Mike. (University of Pennsylvania). *Back Orifice "Remote Administration Tool"* [online]. Available WWW: <URL: http://www.rescomp.upenn.edu/docs/hype/old/bo.html> (1999).

[B64-b1] Glave, James. *Back Orifice a Pain in the...?* [online]. Available WWW: <URL: http://www.wired.com/news/technology/0,1282,14092,00.html> (1998).

[B64-b2] McKay, Niall. *Coming Soon: Back Orifice 2000* [online]. Available WWW: <URL: http://www.wired.com/news/technology/0,1282,20493,00.html> (1999).

[B65-a] Loscocco, Peter A., et al. (National Security Agency). *The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments* [online]. Available WWW: <URL: http://www.jya.com/paperF1.htm> (1998).

[B65-b] Loscocco, Peter A.et al. (National Security Agency). *The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments (Slides)* [online]. Available WWW: <URL: http://www.cs.utah.edu/~sds/inevit-abs.html> (1998).

[B66] CERT/CC. *CERT Advisory CA-95.06* [online]. Available WWW: <URL: http://www.cert.org/advisories/CA-95.06.satan.html> (1995).

[B67] Brackney, Richard. "Cyber-Intrusion Response," 413-415. *Proceedings of the 17th IEEE Symposium on Reliable Distribution Systems.* West Lafayette, IN, Oct. 20-23, 1990. Los Alamitos, CA: IEEE Computer Society Press, 1998.

[B68] Ranum, Marcus. (Network Flight Recorder). "Some Tips on Network Forensics." *Computer Security Institute,* 198 (September 1999): 1-8.

[B69] Cohen, Fred. *Providing for Responsibility in a Global Information Infrastructure* [online]. Available WWW: <URL: http://www.all.net/journal/ntb/responsible.html> (1999).

[B70] Zagar, Mario, et al. *Data Compression Reference Center* [online]. Available WWW: <URL: http://www.rasip.fer.hr/research/compress/index.html> (1999).

[B71] Jensen, Lars Peter & Koch, Peter. (Aalborg University, Denmark). *An Ecological Man-Machine Interface for Temporal Visualization* [online]. Available WWW: <URL: http://www.acm.org/pubs/articles/proceedings/uist/169891/p235-jensen/p235-jensen.pdf> (1992).

[B72] Sawyer, James T.; Minsk, Brian; & Bisantz, Ann M. (Georgia Institute of Technology). *Coupling User Models and System Models: A Modeling Framework for Fault Diagnosis in Complex Systems* [online]. Available WWW: <URL: http://www.eng.buffalo.edu/~bisantz/pubs/um96pap.html> (1996).

[B73] Mitchell, Christine M. (Georgia Institute of Technology). *Models for the Design of Human Interaction with Complex Dynamic Systems* [online]. Available WWW: <URL: http://www.isye.gatech.edu/~cm/papers/model_requirement.10.96.html> (1996).

[B74] Maynard, Terrill D. *Year 2000 Computer Remediation: Assessing Risk Levels in Foreign Outsourcing* [online]. Available WWW: <URL: http://www.SANS.ORG/newlook/resources/Y2K.htm> (1999).

[B75] Rowe, Neil C. & Schiavo, Sandra. *An Intelligent Tutor for Intrusion Detection on Computer Systems* [online]. Available WWW: <URL: http://www.cs.nps.navy.mil/people/faculty/rowe/idtutor.html> (1999).

[B76] Northcutt, Steven. *Network Intrusion Detection.* Indianapolis, IN: New Riders, 1999.

[B77-a] Dockery, Mike & Zajac, John. "Responding to Electronic Evidence Requests." *Electronic Evidence Journal 1,* 1 (October 1, 1996): 1-4 [online]. Available WWW: <URL: http://evidence.finder.com/dockery/FTP/eej10196.pdf>.

[B77-b] Ferraiolo, Karen. (Arca Systems, Inc.). *Tutorial: The Systems Security Engineering Capability Maturity Model* [online]. Available WWW: <URL: http://csrc.nist.gov/nissc/1998/proceedings/tutorB5.pdf> (1998).

[B78] Shumway, Russell M. "Common Sense—An Alternative Approach to Web Security." *Proceedings of the 21st National Information Systems Security Conference.* Arlington, VA, Oct. 5-8, 1998 [online]. Available WWW: <URL: http://csrc.nist.gov/nissc/1998/proceedings/paperD8.pdf>.

[B79] Vaughn, Dr. Rayford B., Jr. (Mississippi State University). "A Practical Approach to Sufficient INFOSEC." *Proceedings of the 21st National Information Systems Security Conference.* Arlington, VA, Oct. 5-8, 1998 [online]. Available WWW: <URL: http://csrc.nist.gov/nissc/1998/proceedings/paperA1.pdf>.

[B80] Ruiu, Dragos. *Cautionary Tales: Stealth Coordinated Attack HOWTO* [online]. Available WWW: <URL: http://www.nswc.navy.mil/ISSEC/CID/Stealth_Coordinated_Attack.html> (1999).

[B81] De Wolf, Hans. *The Jargon File* [online]. Available WWW: <URL: http://web.bilkent.edu.tr/Online/Jargon30/JARGON.HTML> (1999).

[B82] Radcliff, Deborah. *The Danger Within: Internal Employees—Not Outside Hackers—Can Be a Time Bomb Waiting to Blow* [online]. Available WWW: <URL: http://www.idg.net/crd_security_16529.html> (1998).

[B83] Network Associates Technology, Inc. *Next Generation Intrusion Detection in High-Speed Networks* [online]. Available WWW: <URL: http://www.nai.com/media/pdf/nai_labs/ids.pdf> (1999).

[B84] Mitchell, Tom. *Machine Learning.* New York, NY: MacGraw-Hill, 1997.

[B85] Goldberg, David E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. New York, NY: Addison-Wesley, 1989.

[B86] CERT/CC. *CERT Summary CS-99-02* [online]. Available WWW: <URL: http://www.cert.org/summaries/CS-99-02.html> (1999).

[B87] Personal Communication between S. Forrest and J. McHugh.

[B88-1] Hinden, Robert. (Nokia). *IP Next Generation (IPng)* [online]. Available WWW: <URL: http://playground.sun.com/pub/ipng/html/ipng-main.html> (1999).

[B88-2] Deering, Steve & Hinden, Bob. *Statement on IPv6 Address Privacy* [online]. Available WWW: <URL: http://playground.sun.com/pub/ipng/html/ipv6-address-privacy.html> (1999).

[B88-3] Hinden, Robert M. *IP Next Generation Overview* [online]. Available WWW: <URL: http://playground.sun.com/pub/ipng/html/INET-IPng-Paper.html> (1995).

[B89] Amoroso, Edward. *Intrusion Detection*. Sparta, NJ: Intrusion.Net Books, 1999.

[B91] Briney, Andy. *Parker's Plan*. Norwood, MA: Information Security.

[B92] CERT/CC. *CERT Advisory CA-95.06* [online]. Available WWW: <URL: http://www.cert.org/advisories/CA-98.01.smurf.html> (1998).

[B93] PR Newswire Association, Inc. "Plugging the Holes in eCommerce Leads to 135% Growth in the Intrusion Detection and Vulnerability Assessment Software Market," *PRNewswire*. August 10, 1999.

[B94] Computer Security Institute. *3rd Annual CSI/FBI Computer Crime and Security Survey*. March 1998.

[B95] Northcutt, Steven. "Evaluating Intrusion Detection Systems Without Attacking Your Friends," 86. *Network Intrusion Detection*. Indianapolis, IN: New Riders, 1999.

[B96] Stallings, William. *IPv6: The New Internet Protocol* [online]. Available WWW: <URL: http://www.comsoc.org/pubs/surveys/stallings/stallings-orig.html> (1999).

[B97] Arndt, Jonas & Österdahl, Torbjörn. *Network Security in Distributed Systems Using CORBA* [online]. Available WWW: <URL: http://www.etek.chalmers.se/~e3torb/CORBASecurity.pdf> (1998).

[B98] Firth, Robert, et al. *Detecting Signs of Intrusion.* (CMU/SEI-SIM-001). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997 [online]. Available WWW: <URL: http://www.cert.org/security-improvement/modules/m01.html>.

[B99] CERT/CC. *CERT Incident Note IN-99-01 on sscan* [online]. Available WWW: <URL: http://www.cert.org/incident_notes/IN-99-01.html> (1999).

[B100] Sitaker, Kragen. *How to Find Security Holes* [online]. Available WWW: <URL: http://www.dnaco.net/~kragen/security-holes.html> (1999).

[B101] SonOfFire. *Wardialling* [online]. Available WWW: <URL: http://newbie.darkridge.com/logs97/10.31.wardial.txt> (1997).

[B102] Fish & Richardson P.C. *The Emerging Law of Computer Networks—Finding Out What's There: Technical and Legal Aspects of Discovery* [online]. Available WWW: <URL: http://www.fr.com/publis/f_paper21.html> (1998).

[B103] Computers and Law Class. (University at Buffalo School of Law). *Discovery of Computer Data* [online]. Available WWW: <URL: http://wings.buffalo.edu/Complaw/CompLawPapers/printup.html> (1996).

[B104] Swartwood, Dan T. & Heffernan, Richard. *Trends in Intellectual Property Loss, Survey Report* [online]. Available WWW: <URL: http://www.asisonline.org/stat12.html> (1998).

[B105] Gula, Ron. *Broadening the Scope of Penetration Testing Techniques* [online]. Available WWW: <URL: http://www.securityfocus.com/templates/forum_message.html?forum=2&head=7&id=7> (1999).

[B106] Network Security Solutions Ltd. *Techniques Adopted by 'System Crackers' when Attempting to Break into Corporate or Sensitive Private Networks* [online]. Available WWW: <URL: http://www.clark.net/pub/roesch/public_html/cracker.txt> (1998).

[B107] Farmer, Dan & Venema, Wietse. *Improving the Security of Your Site by Breaking Into It* [online]. Available WWW: <URL: http://www.clark.net/pub/roesch/public_html/improve_by_breakin.txt> (1999).

[B108-1] Spitzner, Lance. *Know Your Enemy* [online]. Available WWW: <URL: http://www.enteract.com/~lspitz/enemy.html> (1999).

[B108-2] Spitzner, Lance. *Know Your Enemy: II* [online]. Available WWW: <URL: http://www.enteract.com/~lspitz/enemy2.html> (1999).

[B108-3] Spitzner, Lance. *Know Your Enemy: III* [online]. Available WWW: <URL: http://www.enteract.com/~lspitz/enemy3.html> (1999).

[B108-4] Spitzner, Lance. *How to Build a Honeypot* [online]. Available WWW: <URL: http://www.enteract.com/~lspitz/honeypot.html> (1999).

[B109] Wingfield, Nick. *Java, ActiveX Security Elusive* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1003-200-317102.html> (1997).

[B110] McGraw, Gary. *Java's 2's Verifier Becomes Confused by German Student's Security Attack* [online]. Available WWW: <URL: http://www.javaworld.com/javaworld/jw-04-1999/jw-04-flaw.html> (1999).

[B111] Elgin, Ben. *Risky Business* [online]. Available WWW: <URL: http://www.zdnet.com/devhead/stories/articles/0,4413,1600421,00.html> (1997).

[B112] Coffee, Peter. *Java, ActiveX Under a Microscope* [online]. Available WWW: <URL: http://www.zdnet.com/devhead/stories/articles/0,4413,1600418,00.html> (1996).

[B113] McLain, Fred. *The Exploder Control Frequently Asked Questions* [online]. Available WWW: <URL: http://www.halcyon.com/mclain/ActiveX/Exploder/FAQ.htm> (1997).

[B114] Seminerio, Maria. *Hackers Claim ActiveX Can Be Used to Pilfer Money Online* [online]. Available WWW: <URL: http://www.zdnet.com/devhead/stories/articles/0,4413,1600422,00.html> (1997).

[B115-1] Guttman, Barbara & Bagwill, Robert. *NIST Special Publication—Internet Security Policy: A Technical Guide* [online]. Available WWW: <URL: http://csrc.nist.gov/isptg/> (1997).

[B115-2] Guttman, Barbara & Bagwill, Robert. *NIST Special Publication—Internet Security Policy: A Technical Guide—II* [online]. Available WWW: <URL: http://csrc.nist.gov/isptg/pdf/00CoverPage.pdf> (1997).

[B115-3] Guttman, Barbara & Bagwill, Robert. *NIST Special Publication—Internet Security Policy: A Technical Guide—III* [online]. Available WWW: <URL: http://csrc.nist.gov/isptg/pdf/01Introduction.pdf> (1997).

[B115-4] Guttman, Barbara & Bagwill, Robert. *NIST Special Publication—Internet Security Policy: A Technical Guide—IV* [online]. Available WWW: <URL: http://csrc.nist.gov/isptg/pdf/01TOC.pdf> (1997).

[B115-5] Guttman, Barbara & Bagwill, Robert. *NIST Special Publication—Internet Security Policy: A Technical Guide—V* [online]. Available WWW: <URL: http://csrc.nist.gov/isptg/pdf/02GeneralPolicy.pdf> (1997).

[B116] Kochmar, John, et al. *Preparing to Detect Signs of Intrusion.* (CMU/SEI-SIM-005). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1998 [online]. Available WWW: <URL: http://www.cert.org/security-improvement/modules/m05.html>.

[B117] CERT/CC. *Establish a Policy and Set of Procedures that Prepare Your Organization to Detect Signs of Intrusion* [online]. Available WWW: <URL: http://www.cert.org/security-improvement/practices/p040.html> (1998).

[B118] Vranesevich, John. *How to Become a Hacker Profiler* [online]. Available WWW: <URL: http://www.antionline.com/SpecialReports/profiling-index.html> (1999).

[B119] Kendall, Kristopher. "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems." BS/MS Thesis, Massachusetts Institute of Technology, June 1999.

[B120] Lippman, R.P. et al. "MIT Lincoln Laboratory Offline Component of DARPA 1998 Intrusion Detection Evaluation." *Presentation at MIT Lincoln Laboratory PI Meeting,* December 14, 1998 [online]. Available WWW: <URL: http://ideval.ll.mit.edu/intro-html-dir>.

[B121] Graf, I. et al. "Results of DARPA 1998 Offline Intrusion Detection Evaluation." *Presentation at MIT Lincoln Laboratory PI Meeting,* December 15, 1998 [online]. Available WWW: <URL: http://ideval.ll.mit.edu/results-html-dir>.

[B123] Kossakowski, Peter, et al. *Responding to Intrusions.* (CMU/SEI-SIM-006). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1998 [online]. Available WWW: <URL: http://www.cert.org/security-improvement/modules/m06.html>.

[B124] White, G.; Fisch, E.; & Pooch, U. "Cooperating Security Managers: A Peer-Based Intrusion Detection System." *IEEE Network 10,* 1 (Jan/Feb 1996): 20-23.

[B125] Toure, Maodo. (Université Paul Sabatier). "An Interdisciplinary Approach for Adding Knowledge to Computer Security Systems," 158-168. *Proceedings of the IEEE International Carnahan Conference on Security Technology.* Albuquerque, NM, Oct. 12-14, 1994. New York, NY: IEEE, 1994.

[B126] Proctor, Paul. (SAIC). "Audit Reduction and Misuse Detection in Heterogeneous Environments." *Proceedings of the 10th Annual Computer Security Applications Conference.* Orlando, FL, Dec. 5-9, 1994. Los Alamitos, CA: IEEE Computer Society Press, 1995.

[B127] Escamilla, Terry. *Intrusion Detection: Network Security Beyond the Firewall.* New York, NY: Wiley Computer Publishing, 1998.

[B129] Fyodor. *Nmap—The Network Mapper* [online]. Available WWW: <URL: http://www.insecure.org/nmap/> (1999).

[B130] CERT/CC. *Security for Information Technology Service Contracts.* (CMU/SEI-SIM-003). Available WWW: <URL: http://www.cert.org/security-improvement/modules/m03.html> (1998).

[B131] Cresson-Wood, Charles. *Information Security Policies Made Easy: A Comprehensive Set of Information Security Policies.* Sausalito, CA: Baseline Software, 1997.

[B132] Hart, Rod; Morgan, Darren; & Tran, Hai. (James Madison University). "An Introduction to Automated Intrusion Detection Approaches." *Information Management and Computer Security* 7, 2 (1999): 76-82 [online]. Available WWW: <URL: http://www.emerald-library.com/pdfs/04607bb2.pdf>.

[B133] Brock, Jack L., Jr. (Governmentwide and Defense Information Systems). *NRC's Intrusion Detection and Response Capabilities* (AIMD-99-273R). Washington, DC: United States General Accounting Office, August 1999.

[B134] Power, Richard. "Issues and Trends: 1999 CSI/FBI Computer Crime and Security Survey." *Computer Security Journal XV,* 2 (1999) [online]. Available WWW: <URL: http://www.gocsi.com/losses.htm>.

[B135] Firth, Robert, et al. *An Approach for Selecting and Specifying Tools for Information Survivability.* (CMU/SEI-97-TR-009). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1997.

[B136] Riley, Gary. *CLIPS: A Tool for Building Expert Systems* [online]. Available WWW: <URL: http://www.ghg.net/clips/CLIPS.html> (1999).

[B137] ICSA.net. *About the Intrusion Detection Systems Consortium* [online]. Available WWW: <URL: http://www.icsa.net/html/communities/ids/membership/index.shtml> (1999).

[B138] Check Point Software Technologies, Ltd. *OPSEC Alliance Solutions Center* [online]. Available WWW: <URL: http://www.checkpoint.com/opsec/> (1998).

[B139] Moritz, Ron, et al. *CCIAPI: Common Content Inspection Application Programming Interface* [online]. Available WWW: <URL: www.stardust.com/cciapi/docs/010799/CCIAPIScopeDraft3011.doc> (1999).

[B140] Adaptive Network Security Alliance, Inc. *The Adaptive Network Security Alliance: Industry Leaders Teaming to Imrpove Enterprise Security* [online]. Available WWW: <URL: http://ansa.iss.net/> (1998).

[B142] SEMPER. *IDWG Mail Archive* [online]. Available WWW: <URL: http://www.semper.org/idwg-public> (1999).

[B143] Howard, John D. & Longstaff, Thomas A. *A Common Language for Computer Security Incidents* (SAND98-8667). Albuquerque, NM & Livermore, CA: Sandia National Laboratories, October 1998 [online]. Available WWW: <URL: http://www.cert.org/nav/reports.html>.

[B144] Lippmann, Richard P. et al. "Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation." *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX 2000).* Hilton Head, SC, Jan. 25-27, 2000. Los Alamitos, CA: IEEE Computer Society Press, 2000.

[B145] Lemos, Robert. "ActiveX, Java Holes a Product of Internet Time." Available WWW:<URL http:www.zdnet.com/zdnn.content/0910/zdnn0008.html> (1997).

[B146] Hall, David L. & Llinas, James. "An Introduction to Multisensor Data Fusion." *Proceedings of the IEEE 85*, 1 (January 1997): 6-10.

COMMERCIAL

[C1] Power, Richard & Farrow, Rik. *CSI Intrusion Detection System Resource* [online]. Available WWW: <URL: http://www.gocsi.com/ques.htm> (1998).

[C2-1] Ranum, Marcus J., et al. (Network Flight Recorder, Inc.). *Implementing a Generalized Tool for Network Monitoring* [online]. Available WWW: <URL: http://www.nfr.net/forum/publications/LISA-97.htm> (1997).

[C2-2] Network Flight Recorder, Inc. *The Network Flight Recorder in Action!* [online]. Available WWW: <URL: http://www.nfr.net/products/technology.html> (1997).

[C2-3] Ranum, Marcus J. (Network Flight Recorder, Inc.). *Intrusion Detection: Challenges and Myths* [online]. Available WWW: <URL: http://www.nfr.net/forum/publications/id-myths.html> (1998).

[C3-1] Foote, Steven. (Hurwitz Group). *How Anti-hacker Software Could Have Kept Me Out of Your Company* [online]. Available WWW: <URL: http://www.netect.com/whitepaper.html> (1998).

[C3-2] Netect. *HackerShield—Features and Benefits* [online]. Available WWW: <URL: http://www.netect.com/hs_features.html> (1998).

[C4-1] En Garde Systems, Inc. *En Garde Systems Inc. is Proud to Announce T-sight, the First Advanced Intrusion Investigation and Response Tool for Windows NT* [online]. Available WWW: <URL: http://engarde.com/software/t-sight/overview.html> (1998).

[C4-2] En Garde Systems, Inc. *T-sight: Major Features List* [online]. Available WWW: <URL: http://engarde.com/software/t-sight/features.html> (1998).

[C4-3] En Garde Systems, Inc. *T-sight RealTime—Main Window* [online]. Available WWW: <URL: http://engarde.com/software/t-sight/tutorial/realtime/main.html> (1999).

[C5-1] IBM. *SecureWay FirstSecure* [online]. Available WWW: <URL: http://www.software.ibm.com/security/firstsecure> (1999).

[C5-2] IBM. *IBM Integrated Security Solutions: Comprehensive Security Solutions for Enabling e-business* [online]. Available WWW: <URL: http://www-4.ibm.com/software/security/firstsecure/library/whitepapers/intsecsol.html> (1999).

[C6-1] Mimestar, Inc. *SecureNet PRO Data Sheet* [online]. Available WWW: <URL: http://mimestar.com/html/data_sheet.htm> (1997).

[C6-2] Mimestar, Inc. *SecureNet PRO: The Complete Network Security Solution* [online]. Available WWW: <URL: http://www.mimestar.com/html/products.htm> (1997).

[C6-3] Mimestar, Inc. *SecureNet PRO Frequently Asked Questions* [online]. Available WWW: <URL: http://www.mimestar.com/html/product_faq.htm> (1997).

[C7] Touch Technologies, Inc. *INTOUCH INSA—Network Security Agent* [online]. Available WWW: <URL: http://www.ttisms.com/tti/nsa_www.html> (1996).

[C8] Harris Communications. *Stake Out I.D.* [online]. Available WWW: <URL: http://www.commprod.harris.com/network-security/stakeout/> (1999).

[C9-1] Digital Security InfoCenter. *POLYCENTER Security Intrusion Detector* [online]. Available WWW: <URL: http://www.digital.com/info/security/id.htm> (1996).

[C9-2] Digital. *POLYCENTER Security Intrusion Detector for Digital UNIX, Version 1.2A* [online]. Available FTP: <URL: ftp://ftp.digital.com/pub/Digital/info/SPD/43-08-XX.txt> (1995).

[C10-1] HP OpenView. *Features and Benefits of Node Sentry* [online]. Available WWW: <URL: http://www.openview.hp.com/products/node/features/> (1999).

[C10-2] HP OpenView. *HP OpenView Node Sentry Product Brief* [online]. Available WWW: <URL: http://www.openview.hp.com:80/pdfs/257.pdf> (1999).

[C11] ODS Networks, Inc. *CDMS: Computer Misuse Detection System* [online]. Available WWW: <URL: http://www.ods.com/security/products/cmds.shtml> (1999).

[C12-1] CyberSafe Corporation. *Centrax FAQ's* [online]. Available WWW: <URL: http://www.centraxcorp.com/faq.html> (1999).

[C12-2] CyberSafe Corporation. *Centrax: New Features & Enhancements in Centrax 2.2* [online]. Available WWW: <URL: http://www.centraxcorp.com/centrax22.html> (1999).

[C12-3] CyberSafe Corporation. *Centrax FAQ's* [online]. Available WWW: <URL: http://www.centraxcorp.com/faq.html> (1999).

[C12-4] CyberSafe Corporation. *Centrax Intrusion Detection Software Features and Benefits* [online]. Available WWW: <URL: http://www.centraxcorp.com/benefits.html> (1999).

[C12-5] CyberSafe Corporation. *Centrax* [online]. Available WWW: <URL: http://www.centraxcorp.com/centrax.html> (1999).

[C12-6] CyberSafe Corporation. *Centrax Security Software* [online]. Available WWW: <URL: http://www.centraxcorp.com/zmedia/IRM%20Summer_Fall.pdf> (1999).

[C12-7] CyberSafe Corporation. *Centrax Security Software—II* [online]. Available WWW: <URL: http://www.centraxcorp.com/zmedia/IRM%20Magazine.pdf> (1999).

[C13-1] Van Ryan, Jane. *SAIC's Center for Information Security Technology Releases CMDS Verson 3.5* [online]. Available WWW: <URL: http://www.saic.com/news/may98/news05-15-98.html> (1998).

[C13-2] Proctor, Paul E. (SAIC). *Computer Misuse Detection System (CMDS) Concepts* [online]. Available WWW: <URL: http://cp-its-web04.saic.com/satt.nsf/externalbycat> (1996).

[C14] Net Nanny Software International Inc. *BioPassword: Undeniably Identified—An Overview of Our Patented Keystroke Dynamic Technology* [online]. Available WWW: <URL: http://www.biopassword.com/docs/BioPassword.PDF> (1998).

[C15-a1] Internet Security Systems. *Real Secure* [online]. Available WWW: <URL: http://www.iss.net/prod/realsecure.pdf> (1999).

[C15-a2] Internet Security Systems. *RealSecure System Requirements* [online]. Available WWW: <URL: http://www.iss.net/reqspec/reqDisplay.php3?pageToDisplay=RS%20sys%20reqs> (1999).

[C15-a3] Internet Security Systems. *RealSecure Attack Signatures* [online]. Available WWW: <URL:http://www.iss.net/reqspec/linkDisplay.php3?pageToDisplay=RS%20a.s.%20from%20DB> (1998).

[C15-a4] Internet Security Systems. *Real-Time Attack Recognition and Response: A Solution for Tightening Network Security* [online]. Available WWW: <URL: http://www.iss.net/prod/whitepapers/realtime.pdf> (1999).

[C15-b] Lucent Technologies, Inc. *Network Intrusion Detection in Action* [online]. Available WWW: <URL: http://www.lucent.com/dns/library/pdf/brochures/realsecure.pdf> (1998).

[C16] Computer Associates. *SessionWall-3* [online]. Available WWW: <URL: http://www.abirnet.com/products.html> (1999).

[C17-1] AXENT Technologies, Inc. *NetProwler—Advanced Network Intrusion Detection* [online]. Available WWW: <URL: http://www.axent.com/iti/netprowler/idtk_ds_word_1.html> (1999).

[C17-2] AXENT Technologies, Inc. *Netprowler* [online]. Available WWW: <URL: http://www.axent.com/product/netprowler/default.htm> (1998).

[C17-3] AXENT Technologies, Inc. *Netprowler—II* [online]. Available WWW: <URL: http://www.axent.com/product/netprowler/npbrochure.htm> (1998).

[C18] Security Dynamics. *Kane Security Monitor* [online]. Available WWW: <URL: http://www.securitydynamics.com/products/datasheets/kmds.html> (1999).

[C20-1] Cisco. *NetRanger* [online]. Available WWW: <URL: http://www.cisco.com/warp/public/778/security/netranger/> (1999).

[C20-2] Cisco. *The NetRanger Intrusion Detection System* [online]. Available WWW: <URL: http://www.cisco.com/warp/public/778/security/netranger/prodlit/netra_ov.htm> (1998).

[C20-3] Cisco. *NetRanger Intrusion Detection System* [online]. Available WWW: <URL: http://www.cisco.com/warp/public/778/security/netranger/netra_ds.htm> (1998).

[C20-4] Cisco. *NetRanger—General Concepts* [online]. Available WWW: <URL: http://www.cisco.com/warp/public/778/security/netranger/netra_qp.htm> (1998).

[C21-1] Network Associates, Inc. *CyberCop Monitor* [online]. Available WWW: <URL: http://www.nai.com/asp_set/products/tns/ccmonitor_intro.asp> (1999).

[C21-2] Network Associates, Inc. *CyberCop Scanner* [online]. Available WWW: <URL: http://www.nai.com/asp_set/products/tns/ccscanner_intro.asp> (1999).

[C21-3] Network Associates, Inc. *CyberCop Sting* [online]. Available WWW: <URL: http://www.nai.com/asp_set/products/tns/ccsting_intro.asp> (1999).

[C21-4] Network Associates, Inc. *CyberCop CASL* [online]. Available WWW: <URL: http://www.nai.com/asp_set/products/tns/cccasl_intro.asp> (1999).

[C21-a] Network Associates, Inc. *Next Generation Intrusion Detection in High Speed Networks* [online]. Available WWW: <URL: http://www.nai.com/media/pdf/nai_labs/ids.pdf> (1999).

[C21-b] Network General Corporation. *A Network Visibility Guide—Protecting Your Network: The Choice Between Active and Static Security Technologies* [online]. Available WWW: <URL: http://www.3dg.com/cybercop/ccvg/ccvg1.html> (1997).

[C21-c] Network General Corporation. *CyberCop Datasheet* [online]. Available WWW: <URL: http://www.3dg.com/cybercop/p_s/data1.html> (1997).

[C22-a1] Tripwire Security Systems, Inc. *The History of Tripwire* [online]. Available WWW: <URL: http://www.tripwiresecurity.com/products/history.html> (1998).

[C22-a2] Tripwire Security Systems, Inc. *Tripwire, Inc.: Company Information* [online]. Available WWW: <URL: http://www.tripwiresecurity.com/compintro.html> (1998).

[C22-a3] Tripwire Security Systems, Inc. *Tripwire Academic Source Release 1.3.1* [online]. Available WWW: <URL: http://www.tripwiresecurity.com/products/ASR1_3.html> (1998).

[C22-a4] Tripwire Security Systems, Inc. *Tripwire 2.0 for Unix* [online]. Available WWW: <URL: http://www.tripwiresecurity.com/products/2_0Unix.html> (1998).

[C22-a5] Tripwire Security Systems, Inc. *Tripwire 2.0 for Windows NT* [online]. Available WWW: <URL: http://www.tripwiresecurity.com/products/2_0NT.html> (1998).

[C22-a6] Tripwire Security Systems, Inc. *Tripwire 2.x Enhancements over Tripwire ASR 1.3* [online]. Available WWW: <URL: http://www.tripwiresecurity.com/vs.html> (1998).

[C22-a7] Kim, Gene & McHugh, John. *File Integrity Assessment* [online]. Available WWW: <URL: http://www.tripwiresecurity.com/company/press_releases/webcast.ppt> (1999).

[C22-b] Cohen, Frederick B. *Re: Intrusion Detection, Tripwire, etc.* [online]. Available WWW: <URL: http://www.geek-girl.com/ids/0602.html> (1995).

[C23-a] LOpht Heavy Industries, Inc. *Antisniff—Overview* [online]. Available WWW: <URL: http://www.l0pht.com/antisniff/overview.html> (1999).

[C23-b] Harrison, Ann. *Security Think Tank Releases Sniffer Tool* [online]. Available WWW: <URL: http://www.computerworld.com/home/print.nsf/all/990809BAA2> (1999).

[C24] Roesch, Martin. *The Snort Page* [online]. Available WWW: <URL: http://www.clark.net/~roesch/security.html> (1999).

NEWS ITEMS

[N1] Reuters. *White House Threatens to Punish Hackers* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1005-200-343118.html?tag=st.cn.1> (1999).

[N2] Festa, Paul. (CNET News.com). *Senate, FBI Sites Down on Hack Attacks* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1005-200-343061.html?tag=st.ne.1005-200-343118> (1999).

[N3] Reuters. *Some NASA Systems Easy Prey for Hackers* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1005-200-342779.html?tag=st.ne.1005-200-343061> (1999).

[N4] Reuters. *White House Shuts down Web Site* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1005-200-342364.html?tag=st.ne.1005-200-343118> (1999).

[N5] Shankland, Stephen. (CNET News.com). *U.S. Weapons Labs Shut Down Classified Networks* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1003-200-340847.html?tag=st.ne.1002> (1999).

[N6] Reuters. *NATO Site, Email Suffers Hacks* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1005-200-340625.html?tag=st.ne.1> (1999).

[N7] Festa, Paul. (CNET News.com). *Defense Department Fights off Hackers* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1005-200-339584.html?tag=st.ne.1005-200-343118> (1999).

[N8] Clark, Tim. (CNET News.com). *Navy Fights New Hack* [online]. Available WWW: <URL: http://news.cnet.com/news/0-1003-200-333601.html?tag=st.cn.1> (1998).

[N9] "Cyber-theft of Sensitive U.S. Files Traced to Russia." *Chicago Sun-Times.* October 7, 1999.

[N10] Verton, Daniel. *Cyberattacks Against DOD up 300 Percent this Year* [online]. Available WWW: <URL: http://www.fcw.com/fcw/articles/fcw_11031999_attack.asp> (1999).

[N11] Kerber, Ross. (*The Boston Globe*). *A Handle on Hackers* [online]. Available WWW: <URL: http://www.boston.com/dailyglobe2/335/business/Handle_on_hackers%2b.shtml> (1999).

[N12] Yasin, Rutrell. *Rise in Instrusions Sparks Concern* [online]. Available WWW: <URL: http://www.internetwk.com/story/INW19991130S0007> (1999).

RESEARCH

[R1-a] Lunt, Teresa F. (SRI International). *Detecting Intruders in Computer System* [online]. Available WWW: <URL: http://www2.csl.sri.com/nides/index5.html> (1993).

[R1-b] Lunt, Teresa F., et al. (SRI International). *A Real-Time Intrusion Detection Expert System (IDES)* [online]. Available WWW: <URL: http://www2.csl.sri.com/nides.index5.html> (1992).

[R1-c] Anderson, Debra; Frivold, Thane; & Valdes, Alfonso. (SRI International). *Next-Generation Intrusion Detection Expert System (NIDES), A Summary* (SRI-CSL-95-07). Menlo Park, CA: Computer Science Laboratory, SRI International, May 1995 [online]. Available WWW: <URL: http://www.sdl.sri.com/nides.index5.html>.

[R1-d] Anderson, Debra, et al. (SRI International). *Detecting Unusual Program Behavior Using the Statistical Component of the Next-Generation Intrusion Detection Expert System (NIDES)* (SRI-CSL-95-06). Menlo Park, CA: Computer Science Laboratory, SRI International, May 1995 [online]. Available WWW: <URL: http://www.sdl.sri.com/nides/index5.html>.

[R2-a] Porras, Phillip A. Neumann, Peter G. (SRI International). *EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances* [online]. Available WWW: <URL: http://www2.csl.sri.com/emerald/concepts.html> (1999).

[R2-b] Porras, Phillip A. & Neumann, Peter G. (SRI International). *EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances* [online]. Available WWW: <URL: http://www2.csl.sri.com/emerald/presentations/NISSC97/sld001.htm> (1997).

[R2-c] Computer Science Laboratory. (SRI International). *History of Intrusion Detection at SRI/CSL* [online]. Available WWW: <URL: http://www2.csl.sri.com/intrusion/intrusion-main.html> (1997).

[R2-d] Porras, Phillip A. & Neumann, Peter G. (SRI International). *EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances* [online]. Available WWW: <URL: http://www2.csl.sri.com/emerald/downloads.html> (1998).

[R2-e] Porras, Phillip A. & Valdes, Alfonso. (SRI International). "Live Traffic Analysis of TCP/IP Gateways." *Proceedings of the 1998 Internet Society Symposium on Network and Distributed System Securty.* March 1998 [online]. Available WWW: <URL: http://www.sdl.sri.com/emerald/downloads.html>.

[R2-f] Neumann, Peter G. & Porras, Phillip A. (SRI International). "Experience with EMERALD to Date." *Proceedings of the 1st USENIX Workshop on Intrusion Detection and Network Monitoring.* Santa Clara, CA, Apr. 11-12, 1999 [online]. Available WWW: <URL: http://www.sdl.sri.com/emerald/downloads.html>.

[R3] Kemmerer, Richard A. (University of California, Santa Barbara). *NSTAT: A Model-Based Real-Time Network Intrusion Detection System* (TRCS97-18). November 1997 [online]. Available WWW: <URL: http://www.cs.ucsb.edu/~kemm/netstat.html/documents.html> .

[R4] Kemmerer, Richard A., et al. (University of California, Santa Barbara). *STAT Projects* [online]. Available WWW: <URL: http://www.cs.ucsb.edu/~kemm/netstat.html/projects.html> (1999).

[R5] Ilgun, Koral; Kemmerer, Richard A.; & Porras, Phillip A. (University of California, Santa Barbara). "State Transition Analysis: A Rule-Based Intrusion Detection Approach." *IEEE Transactions on Software Engineering Vol. 21, #3 (SE-21)* (March 1995): 1-22 [online]. Available WWW: <URL: http://www.cs.ucsb.edu/~kemm/nestat.html/documents/html>.

[R6-a] Bass, Tim. (The Silk Road Group Ltd). *Intrusion Detection Systems & Multisensor Data Fusion: Creating Cyberspace Situational Awareness—Introduction* [online]. Available WWW: <URL: http://www.silkroad.com/paper/html/ids/node1.html> (1999).

[R6-b] Bass, Tim. (The Silk Road Group Ltd). "Multisensor Data Fusion for Next Generation Distributed Intrusion Detection Systems." *Proceedings of the 1999 IRIS National Symposium on Sensor and Data Fusion.* May 24-27, 1999 [online]. Available WWW: <URL: http://www.silkroad.com/papers/html/iris>.

[R6-c] Bass, Tim & Gruber, Dave. "A Glimpse into the Future of ID." *;login: The USENIX Association Magazine* (July 1999) [online]. Available WWW: <URL: http://www.silkroad.com/papers/html/glimpse>.

[R7] Kumar, Sandeep & Spafford, Eugene. H. (Purdue University). *An Application of Pattern Matching in Intrusion Detection* (CSD-TR-94-013). West Lafayette, IN: COAST Laboratory, Purdue University, 1994 [online]. Available WWW: <URL: http://www.cs.purdue.edu/coast/coast-library.html>.

[R8] Kumar, Sandeep & Spafford, Eugene H. (Purdue University). *A Pattern Matching Model for Misuse Intrusion Detection* (Coast TR 95-06) [online]. Available WWW: <URL: http://www.cs.purdue.edu/coast/coast-library.html> (1994).

[R9] Porras, Phil, et al. (University of California, Davis). *The Common Intrusion Detection Framework Architecture* [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/cidf/> (1999).

[R10] Sanchez, Luis A.; Kent, Stephen T.; & DiBlasio, Marguerite I. (BBN Systems and Technologies). *External Routing Intrusion Detection Systems (ERIDS)—1998 Project Summary* [online]. Available WWW: <URL: http://www.darpa.mil/ito/psum1998/G403-0.html> (1998).

[R11] Cheung, Steven, et al. (University of California, Davis). *The Design of GrIDS: A Graph-Based Intrusion Detection System* (CSE-99-2). Davis, CA: Department of Computer Science, University of California at Davis, 1999 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/papers.html>.

[R12-a] Staniford-Chen, S., et al. (University of California, Davis). "GrIDS—A Graph-Based Intrusion Detection System for Large Networks." *Proceedings of the 19th National Information Systems Security Conference* [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/papers.html> (1996).

[R12-b] Levitt, Karl, et al. (University of California, Davis). *GrIDS Requirements Document* [online]. Available WWW: <URL: http://olympus.cs.ucdavis.edu/arpa/grids/requirements.html> (1999).

[R13] Loyall, Joe. (GTE). "Toolkit for Creating Adaptable Distributed Applications." *Proceedings of the DARPA Intrusion Detection Meeting.* Dec. 15-17, 1998 [online]. Available WWW: <URL: http://www.dist-systems.bbn.com/projects/OIT>.

[R14] Levitt, Karl N. (University of California, Davis). "Global Guard." *Proceedings of the DARPA Intrusion Detection PI Meeting.* Lexington, MA, Dec. 15-17, 1998 [online]. Available WWW: <URL: http://www.dyncorp-is.com/darpa/meetings/id98dec/agenda.html>.

[R15] Stolfo, Sal. (Columbia University). "The JAM Project & Evaluation Update." *Proceedings of the DARPA Intrusion Detection PI Meeting.* Lexington, MA, Dec. 15-17, 1998 [online]. Available WWW: <URL: http://www.dyncorp-is.com/darpa/meetings/id98dec/agenda.html>.

[R16] Lee, Wenke & Stolfo, Salvatore J. (Columbia University). *Data Mining Approaches for Intrusion Detection* [online]. Available WWW: <URL: http://www.cs.columbia.edu/~sal/hpapers/USENIX/usenix.html> (1999).

[R17] Lee, Wenke; Stolfo, Salvatore J.; & Chan, Philip K. "Learning Patterns from Unix Process Execution Traces for Intrusion Detection" *Proceedings of AAAI Workshop: AI approaches to Fraud Detection and Risk Management.* AAAI Press, July 1997.

[R18] Cohen, William W. (AT&T Laboratories). *Learning Trees and Rules with Set-Valued Features* [online]. Available WWW: <URL: http://www.research.att.com/~wcohen/> (1996).

[R19] Cohen, William W. (AT&T Bell Laboratories). "Fast Effective Rule Induction." *Proceedings of the 12th International Conference on Machine Learning.* Lake Tahoe, CA, 1995 [online]. Available WWW: <URL: http://www.research.att.com/~wcohen/>.

[R20] Heberlein, L. Todd, et al. (University of California, Davis). "A Network Security Monitor," 296-304. *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy.* Oakland, CA, May 7-9, 1990 [online]. Available WWW: <URL: http://olympus.cs.ucdavis.edu/papers.html>.

[R21] Snapp, Steven R., et al. (University of California, Davis). "DIDS (Distributed Intrusion Detection System)—Motivation, Architecture, and an Early Prototype," 167-176. *Proceedings of the 14th National Computer Security Conference.* Washington, DC, Oct. 1991 [online]. Available WWW: <URL: http://olympus.cs.ucdavis.edu/papers.html>.

[R22-1] Crosbie, Mark & Spafford, Gene. (Purdue University). *Active Defense of a Computer System Using Autonomous Agents.* (CSD-TR-95-008) West Lafayette, IN: COAST Laboratory, Purdue University, 1995 [online]. Available WWW: <URL: http://www.cs.purdue.edu/coast/coast-library.html>.

[R22-2] Crosbie, Mark & Spafford, Gene. (Purdue University). "Applying Genetic Programming to Intrusion Detection." *Proceedings of the AAAI Fall Symposium on Genetic Programming.* Cambridge, MA, Nov. 10-12, 1995. Menlo Park, CA: AAAI Press, 1995.

[R23] Garvey, T.; & Lunt, T. (SRI International). "Model-Based Intrusion Detection," 374-385. *Proceedings of the 14th National Computer Security Conference.* Washington, DC, Oct. 1991.

[R24-a] Lawrence Livermore National Laboratory Computer Security Technology Center. *NID Introduction* [online]. Available WWW: <URL: http://ciac.llnl.gov/cstc/nid/intro.html> (1998).

[R24-b] Lawrence Livermore National Laboratory Computer Security Technology Center. *NID Distribution Site* [online]. Available WWW: <URL: http://ciac.llnl.gov/cstc/nid/nid.html> (1999).

[R25-1] Maxion, Roy A. (Carnegie Mellon University). *Invictus: Detection of Unanticipated Anamalies in Evolutionary Environments* [online]. Available WWW: <URL: http:// www.cs.cmu.edu/~maxion/invictus> (1999).

[R25-2] Maxion, Roy A. (Carnegie Mellon University). *Cinnamon: Synthetic Data Generation* [online]. Available WWW: <URL: http://www.cs.cmu.edu/~maxion/invictus/cinnamon.html> (1999).

[R25-3] Maxion, Roy A. (Carnegie Mellon University). *Harbinger: Anomaly Detection Techniques* [online]. Available WWW: <URL: http://www.cs.cmu.edu/~maxion/invictus/ harbinger.html> (1999).

[R25-4] Maxion, Roy A. (Carnegie Mellon University). *Invictus: Toward Dependable Systems* [online]. Available WWW: <URL: http:// www.cs.cmu.edu/~maxion/invictus/InvQuad.jpg> (1999).

[R27] Mé, Ludovic. (Supélec). *Genetic Algorithms, an Alternative Tool for Security Audit Trail Analysis* [online]. Available WWW: <URL: http://www.supelec-rennes.fr/rennes/si/equipe/lme/these/ these-lm.html> (1995).

[R28] Teng, Henry S.; Chen, Kaihu; & Lu, Stephen C-Y. "Security Audit Trail Analysis using Inductively Generated Predictive Rules," 24-29 vol.1. *Proceedings of the 6th Conference on Artificial Intelligence Applications.* Santa Barbara, CA, May 5-9, 1990. Los Alamitos, CA: IEEE Computer Society Press, 1990.

[R29] Frincke, D., et al. (University of Idaho). *A Framework for Cooperative Intrusion Detection* [online]. Available WWW: <URL: http://www.csds.uidaho.edu/~hummer/html/papers.html>.

[R30] Vigna, Giovanni & Kemmerer, Richard A. (University of California, Santa Barbara). "NetSTAT: A Network-Based Intrusion Detection Approach." *Proceedings of the 14th Annual Computer Security Applications Conference.* Scottsdale, AZ, Dec. 1998 [online]. Available WWW: <URL: http://www.cs.ucsb.edu/~kemm/netstat.html/documents.html>.

[R31] Paxson, Vern. (Lawrence Berkeley National Laboratory). "Bro: A System for Detecting Network Intruders in Real-Time," *Proceedings of 7th USENIX Security Symposium.* San Antonio, TX, January 1998 [online]. Available WWW: <URL: http://www.aciri.org/vern/papers.html>.

[R32] Kuykendall, David R. *DIDS—Re: Intro; Question* [online]. Available WWW: <URL: http://www.geek-girl.com/ids/0790.html> (1996).

[R33] Lindqvist, Ulf & Porras, Phillip A. "Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST)." *Proceedings of the 1999 IEEE Symposium on Security and Privacy.* Oakland, CA, May 9-12, 1999 [online]. Available WWW: <URL: http://www2.csl.sri.com/emerald/pbest-sp99-cr.pdf>.

[R34] Bradley, Kirk A., et al. (University of California, Davis). *Detecting Disruptive Routers: A Distributed Network Monitoring Approach* [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/papers/oakland98-paper.pdf> (1998).

[R35] Irwin, Vicki; Northcutt, Stephen; & Ralph, Bill. (Naval Surface Warfare Center). *Building a Network Monitoring and Analysis Capability*—Step by Step [online]. Available WWW: <URL: http://www.nswc.navy.mil/ISSEC/CID/step.htm> (1998).

[R37] Floyd, Sally, et al. (Lawrence Berkeley National Laboratory). *LBNL's Network Research Group* [online]. Available FTP: <URL: http://ftp.ee.lbl.gov/> (1998).

[R38] Network Flight Recorder, Inc. *Step-by-Step Network Monitoring Using NFR* [online]. Available WWW: <URL: http:// www.nswc.navy.mil/ISSEC/CID/nfr.htm> (1998).

[R39] Stocksdale, Greg. *CIDER Documents* [online]. Available WWW: <URL: http://www.nswc.navy.mil/ISSEC/CID/> (1999).

[R40-1] Baccala, Brent. "TCPdump." *Connected: An Internet Encyclopedia,* 3rd ed [online]. Available WWW: <URL: http:// www.freesoft.org/CIE/Topics/55.htm>.

[R40-2] Baccala, Brent. "TCPdump(1)." *Connected: An Internet Encyclopedia,* 3rd ed [online]. Available WWW: <URL: http:// www.freesoft.org/CIE/Topics/56.htm>.

[R41] White, Gregory B.; Fisch, Eric A.; & Pooch, Udo W. "Cooperating Security Managers: A Peer-Based Intrusion Detection System." *IEEE Network* (January/February 1996): 20-23.

[R42] Eliot, Lance B. (Eliot & Associates). "Typing your ID via AI." *AI Expert* (January 1995):9-10.

[R43] Toure, Maodo. (Université Paul Sabatier). "An Interdisciplinary Approach for Adding Knowledge to Computer Security Systems," 158-168. *Proceedings of the IEEE International Carnahan Conference on Security Technology.* Albuquerque, NM, Oct. 12-14, 1994. New York, NY: IEEE, 1994.

[R44] Safford, Dave; Schales, Doug; & Hess, Dave. (Texas A&M University). *Texas A&M Network Security Package Overview* [online]. Available FTP: <URL: ftp://coast.cs.purdue.edu/pub/tools/unix/netlog/TAMU/OVERVIEW> (1993).

[R45] Proctor, Paul. (SAIC). "Audit Reduction and Misuse Detection in Heterogeneous Environments." *Proceedings of the 10th Annual Computer Security Applications Conference.* Orlando, FL, Dec. 5-9, 1994. Los Alamitos, CA: IEEE Computer Society Press, 1995.

[R46-a] Bonifácio, J. M., Jr., et al. "An Adaptive Intrusion Detection System Using Neural Networks." *Proceedings of the IFIP World Computer Congress—Security in Information Systems (IFIP-SEC '98).* Viena, Austria, August/September 1998 [online]. Available WWW: <URL: http://www. icmsc.sc.usp.br/~andre/papers.html>.

[R46-b] Bonifácio, José Maurício, Jr., et al. "Neural Networks Applied in Intrusion Detection Systems." *Proceedings of the IEEE World Congress on Computational Intelligence (WCCI '98).* Anchorage, AK, May 1998 [online]. Available WWW: <URL: http://www.icmsc.sc.usp.br/~andre/papers.html>.

[R47] Forrest, Stephanie; Hofmeyr, Steven A.; & Somayaji, Anil (University of New Mexico). "Computer Immunology." *Communications of the ACM 40,* 10 (1997): 86-96 [online]. Available WWW: <URL: http://www.cs.unm.edu/~forrest/papers.html>.

[R48] Forrest, Stephanie, et al. "A Sense of Self for Unix Processes," 120-128. *Proceedings of the 1996 IEEE Symposium on Security and Privacy.* Los Alamitos, CA: IEEE Computer Society Press, 1996 [online]. Available WWW: <URL: http://www.cs.unm.edu/~forrest/papers.html>.

[R49] D'haeseleer, Patrik; Forrest, Stephanie; & Helman, Paul. (University of New Mexico). *A Distributed Approach to Anomaly Detection* [online]. Available WWW: <URL: http://www.cs.unm.edu/~forrest/papers.html> (1997).

[R50] Somayaji, Anil; Hofmeyr, Steven; &Forrest, Stephanie. (University of New Mexico). "Principles of a Computer Immune System," 75-82. *Proceedings of the 1997 New Security Paradigms Workshop.* 1998 [online]. Available WWW: <URL: http://www.cs.unm.edu/~forrest/papers.html>.

[R51-a] Goan, Terrance. (Stottler Henke Associates, Inc.). "A Cop on the Beat: Collecting and Appraising Intrusion Evidence." *Communications of the ACM 42*, 7 (July 1999): 46-52.

[R51-b] Goan, Terrance. (Stottler Henke Associates, Inc.). *ICE: Intelligent Correlation of Evidence for Intrusion Detection* (183).

[R52-a] University of Idaho. *Hummer Project Intrusion Detection System* [online]. Available WWW: <URL: http://www.csds.uidaho.edu/~hummer/home.html> (1999).

[R52-b] Evans, Jason & Frincke, Deborah. (University of Idaho). *Trust Mechanisms for Hummingbird* [online]. Available WWW: <URL: www1.acm.org/crossroads/xrds2-4/humming.html> (1996).

[R53-1] Stolfo, Salvatore J.; Backenroth, Adam; & Chan, Phil. *The JAM Project* [online]. Available WWW: <URL: http://www.cs.columbia.edu/~sal/JAM/PROJECT/> (1999).

[R53-2] Stolfo, Salvatore J. (Columbia University). *Fraud and Intrusion Detection for Financial Information Systems* [online]. Available WWW: <URL: http://www.cs.columbia.edu/~sal/JAM/PROJECT/EYR1997.html> (1998).

[R54-1] Spafford, Gene, et al. (Purdue University). *Autonomous Agents for Intrusion Detection* [online]. Available WWW: <URL: http://www.cs.purdue.edu/coast/projects/autonomous-agents.html> (1999).

[R54-2] Spafford, Eugene & Zamboni, Diego (Purdue University). *Release of the Alpha Version of the AAFID Prototype* [online]. Available WWW: <URL: http://www.cs.purdue.edu/coast/projects/aafid-announce.html> (1998).

[R54-3] Balasubramaniyan, Jai, et al. (Purdue University). *An Architecture for Intrusion Detection Using Autonomous Agents* (Coast TR 98-05). West Lafayette, IN: COAST Laboratory, Purdue University, 1998 [online]. Available WWW: <URL: http://www.cs.purdue.edu/coast/projects/autonomous-agents.html>.

[R55] Lane, Terran & Brodley, Carla E. (Purdue University). "Temporal Sequence Learning and Data Reduction for Anomaly Detection." *Proceedings of the 5th Conference on Computer and Communications Security.* San Francisco, CA [online]. Available WWW: <URL: http://www.acm.org/pubs/articles/proceedings/commsec/288090/p150-lane/p150-lane.pdf> (1998).

[R56] Jordan, Sabina E., et al. "Discrete-Event Simulation for the Design and Evaluation of Physical Protection Systems." *Proceedings of the 1998 Winter Simulation Conference* [online]. Available WWW: <URL: http://www.acm.org/pubs/articles/proceedings/simulation/293172/p899-jordan/p899-jordan.pdf> (1998).

[R57-1] Cohen, Fred. *Distributed Coordinated Attacks—Background* [online]. Available WWW: <URL: http://all.net/books/dca/background.html> (1996).

[R57-2] Cohen, Fred. *DCA's—A Class of Attacks* [online]. Available WWW: <URL: http://all.net/books/dca/class.html> (1996).

[R57-3] Cohen, Fred. *Characteristics of DCAs* [online]. Available WWW: <URL: http://all.net/books/dca/character.html> (1996).

[R57-4] Cohen, Fred. *Defenses Against DCAs* [online]. Available WWW: <URL: http://all.net/books/dca/defenses.html> (1996).

[R57-5] Cohen, Fred. *A Mathematical Characterization of DCAs* [online]. Available WWW: <URL: http://all.net/books/dca/math.html> (1996).

[R57-6] Cohen, Fred. *Distributed Coordinated Attacks—Summary, Conclusions, and Further Work* [online]. Available WWW: <URL: http://all.net/books/dca/summary.html> (1996).

[R58] Cohen, Fred. *Simulating Cyber Attacks, Defenses, and Consequences* [online]. Available WWW: <URL: http://all.net/journal/ntb/simulate/simulate.html> (1999).

[R59] Moran, Douglas B. (SRI International). *Future Directions for Intrusion Detection* [online]. Available WWW: <URL: http://www.ai.sri.com/~debri/presentations/idwk9507/idwk9507.html> (1996).

[R60] Stillerman, Matthew; Marceau, Carla; & Stillman, Maureen. (Oddyssey Research Associates). "Intrusion Detection for Distributed Applications." *Communications of the ACM 42,* 7 (July 1999): 62-69.

[R61] Ghosh, Anup K.; Wanken, James; & Charron, Frank. (Reliable Software Technologies). "Detecting Anomalous and Unknown Intrusions Against Programs," 259-267. *Proceedings of the 14th Annual Computer Security Applications Conference.* Phoenix, AZ, Dec. 7-11, 1998. Los Alamitos, CA: IEEE Computer Society Press, 1999.

[R62] Helmer, Guy G., et al. "Intelligent Agents for Intrusion Detection," 121-124. *Proceedings of the 1998 IEEE Information Technology Conference, Environment for the Future.* Syracuse, NY, Sept. 1-3, 1998. New York, NY: IEEE, 1998.

[R63] Ye, Nong; Giordano, Joseph; Feldman, John; & Zhong, Qiu. "Information Fusion Techniques for Network Intrusion Detection," 117-120. *Proceedings of 1998 IEEE Information Technology Conference, Environment for the Future.* Syracuse, NY, Sept. 1-3, 1998. New York, NY: IEEE, 1998

[R64] Vert, Greg; Frinke, Deborah A.; & McConnell, Jesse C. (University of Idaho). *A Visual Mathematical Model for Intrusion Detection* [online]. Available WWW: <URL: http://www.csds.uidaho.edu/~hummer/html/papers.html> (1998).

[R65] Ho, Yuan; Frinke, Deborah; & Tobin, Donald, Jr. (University of Idaho). *Planning, Petri Nets, and Intrusion Detection* [online]. Available WWW: <URL: http://www.csds.uidaho.edu/~hummer/html/papers.html> (1998).

[R66] Frinke, Deborah, et al. (University of Idaho). *Research Issues in Cooperative Intrusion Detection Between Multiple Domains* [online]. Available WWW: <URL: http://www.csds.uidaho.edu/~hummer/html/papers.html>.

[R67] Hofmeyr, Steven A.; Forrest, Stephanie; & Somayaji, Anil. (University of New Mexico). "Intrusion Detection Using Sequences of System Calls." *Journal of Computer Society 6*, 3 (1998): 151-180.

[R68-a] Spafford, Gene, et al. (Purdue University). *Audit Trail Reduction* [online]. Available WWW: <URL: http://www.cs.purdue.edu/coast/projects/audit-trails-reduce.html> (1999).

[R68-b] Spafford, Gene, et al. (Purdue University). *Audit Trails Format* [online]. Available WWW: <URL: http://www.cs.purdue.edu/coast/projects/audit-trails-format.html> (1998).

[R69] Braden, Bob. (ISI). *NNStat* [online]. Available WWW: <URL: http://www.duth.gr/InfoBase/noc/nnstat.txt> (1993).

[R70] Cannady, James. (Nova Southeastern University). "Artificial Neural Networks for Misuse Detection." *Proceedings of the 21st National Information Systems Security Conference.* Arlington, VA, Oct. 5-8, 1998 [online]. Available WWW: <URL: http://csrc.nist.gov/nissc/1998/proceedings/paperF13.pdf>.

[R71] Krsul, Ivan; Spafford, Eugene; & Tripunitata, Mahesh. (Purdue University). "An Analysis of Some Software Vulnerabilities." *Proceedings of the 21st National Information Systems Security Conference.* Arlington, VA, Oct. 5-8, 1998 [online]. Available WWW: <URL: http://csrc.nist.gov/nissc/1998/proceedings/paperD6.pdf>.

[R72] Stutz, John & Cheeseman, Peter. (NASA Ames Research Center). *A Short Exposition on Bayesian Inference and Probability* [online]. Available WWW: <URL: http://ic.arc.nasa.gov/ic/projects/bayes-group/html/bayes-theorem-long.html> (1994).

[R73] Varshney, P. *Distributed Detection and Data Fusion.* New York, NY: Springer Verlag, 1996.

[R74] Ilgun, Koral; Kemmerer, Richard A.; & Porras, Phillip A. "State-Transition Analysis: A Rule-Based Intrusion Detection Approach." *IEEE Transactions on Software Engineering XX,* Y (1995): 1-20 [online]. Available WWW: <URL: http://www.cs.ucsb.edu/~kemm/nestat.html/documents.html>.

[R75] Höglund, Albert. (Nokia Research Center). *A UNIX Anomaly Detection System Using Self-Organising Maps* [online]. Available WWW: <URL: http://www.zurich.ibm.com/pub/Other/RAID/Prog_RAID98/Full_Papers/hoglund_slides.html/index.htm> (1998).

[R77-a] Axelsson, Stefan. (Chalmers University, Sweden). "The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection." *Proceedings of the 6th ACM Conference on Computer and Communications Security,* Kent Ridge Digital Labs, Singapore, Nov. 1-4, 1999. ACM, 1999.

[R77-b] Axelsson, Stefan. (Chalmers University, Sweden). "On a Difficulty of Intrusion Detection." *Proceedings of the 2nd International Workshop on Recent Advances in Intrusion Detection (RAID '99).* West Lafayette, IN, Sept. 7-9, 1999.

[R78] McGraw, Gary. (Reliable Software Technologies). "Why Monitoring Mobile Code is Harder than It Sounds." *;login: The USENIX Association Magazine* (September 1999): 18-20.

[R79] Amoroso, Edward. (AT&T Labs). "Design and Integration Principles for Large Scale Infrastructure Protection." *;login: The USENIX Association Magazine* (September 1999): 20-21.

[R80] Yuill, Jim, et al. "Intrusion Detection for an On-Going Attack." *Proceedings of the 2nd International Workshop on Recent Advances in Intrusion Detection (RAID '99).* West Lafayette, IN, Sept. 7-9, 1999 [online]. Available WWW: <URL: http://www.cerias.purdue.edu/raidprog.html>.

[R81] Mansfield, Glenn, et al. "Towards Trapping Wily Intruders in the Large." *Proceedings of the 2nd International Workshop on Recent Advances in Intrusion Detection (RAID '99).* West Lafayette, IN, Sept. 7-9, 1999 [online]. Available WWW: <URL: http://www.cerias.purdue.edu/raidprog.html>.

[R82] Bishop, Matt. (University of California, Davis). "Vulnerabilities Analysis: Extended Abstract." *Proceedings of the 2nd International Workshop on Recent Advances in Intrusion Detection (RAID '99).* West Lafayette, IN, Sept. 7-9, 1999 [online]. Available WWW: <URL: http://www.cerias.purdue.edu/raidprog.html>.

[R83-a] Christey, Stephen; Mann, David; & Hill, William. (The MITRE Corporation). "The Development of a Common Vulnerability Enumeration." *Proceedings of the 2nd International Workshop on Recent Advances in Intrusion Detection (RAID '99).* West Lafayette, IN, Sept. 7-9, 1999 [online]. Available WWW: <URL: http://www.cerias.purdue.edu/raidprog.html>.

[R83-b] Mann, David E. & Christey, Steven M. (The MITRE Corporation). "Towards a Common Enumeration of Vulnerabilities." *Second Workshop on Research with Security Vulnerability Databases.* West Lafayette, IN, Jan. 21-22, 1999 [online]. Available WWW: <URL: http://cve.mitre.org/docs/cerias.html>.

[R83-c] Christey, Steven M. *Re: IMPORTANT: CVE—Common Vulnerabilities and Exposures?* [online]. Available WWW: <URL: http://cve.mitre.org/archives/msg00430.html> (1999).

[R84] Paxton, Vern. (Lawrence Berkeley National Labs). "Experiences Learned from Bro." *;login: The USENIX Association Magazine* (September 1999): 21-22.

[R85] Heberlein, Todd. (Net Squared). *Anomaly Detection of Misuse Reports*[online]. Available WWW: <URL: http://www.netsq.com/Information/RTID-Mining> (1999).

[R86] Brumley, David. (Stanford University). "Invisible Intruders: Rootkits in Practice." *;login: The USENIX Association Magazine* (September 1999): 27-29.

[R87] Brutch, Paul C.; Brutch, Tasneem G.; & Pooch, Udo. (Texas A&M). "Indicators of UNIX Host Compromise." *;login: The USENIX Association Magazine* (September 1999): 30-35.

[R88] Mudge. "A Hacker's Approach to ID." *;login: The USENIX Association Magazine* (September 1999): 36-39.

[R89] Sellens, John. (GNAC Canada). "On Reliability." *;login: The USENIX Association Magazine* (September 1999): 46-52.

[R90] Seleznyov, Alexandr & Puuronen, Seppo. (University of Jyväskylä). "Anomaly Intrusion Detection Systems: Handling Temporal Relations Between Events." *Proceedings of the 2nd International Workshop on Recent Advances in Intrusion Detection (RAID '99).* West Lafayette, IN, Sept. 7-9, 1999 [online]. Available WWW: <URL: http://www.cerias.purdue.edu/raidprog.html>.

[R91] Wespi, A.; Dacier, M. & Debar, H. (IBM Zurich). *Intrusion Detection Using Variable-Length Audit Trail Patterns* (RZ 3164). Zurich, Switzerland: IBM Research, 1999 [online]. Available WWW: <URL: http://domino.watson.ibm.com/library/CYBERDIG.NSF/95f0a8c5802d9417852566a90057461f/02a4ec9d5b79ae14852567da0034838f/$FILE/rz3164.ps>.

[R92] Graf, Isaac, et al. (MIT Lincoln Laboratory). "Results of DARPA 1998 Offline Intrusion Detection Evaluation." *Proceedings of the Intrusion Detection PI Meeting.* Lexington, MA, Dec. 15-17, 1998 [online]. Available WWW: <URL: http://www.dyncorp-is.com/darpa/meetings/id98dec/files/mit-ll.pdf>.

[R93] Maxion, Roy A. (Carnegie Mellon University). "Measuring Intrusion Detection Systems." *Proceedings of the 1st International Workshop on Recent Advances in Intrusion Detection (RAID-98).* Louvain-la-Neuve, Belgium, Sept. 14-16, 1998 [online]. Available WWW: <URL: http://www.zurich.ibm.com/pub/Other/RAID/Prog_RAID98/Full_Papers/maxion.pdf>.

[R94] Koob, Gary. (DARPA/ITO). "Research Challenges to Operating System Security." *Proceedings of the DARPA/NSA Workshop on Operating System Security.* May 22-23, 1996 [online]. Available WWW: <URL: http://www.arpa.mil/ito/Proceedings/OS_Security/challenges/challenges_long.html>.

[R96] Ko, Calvin; Fink, George; & Levitt, Karl. (University of California, Davis). *Automated Detection of Vulnerabilities in Privileged Programs by Execution Monitoring* [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/papers/kfl94.ps> (1999).

SURVEYS

[S1] Aslam, Taimur; Krsul, Ivan; & Spafford, Eugene H. (Purdue University). *Use of a Taxonomy of Security Faults* (Coast TR-96-051). West Lafayette, IN: COAST Laboratory, Purdue University, 1996 [online]. Available WWW: <URL: http://www.cs.purdue.edu/coast/coast-library.html>.

[S2] Frank, Jeremy. (University of California, Davis). *Artificial Intelligence and Intrusion Detection: Current and Future Directions* [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/papers/ncsc.94.ps> (1994).

[S3] Sundaram, Aurobindo. *An Introduction to Intrusion Detection* [online]. Available WWW: <URL: http://www1.acm.org/crossroads/xrds2-4/intrus.html> (1996).

[S4] Herringshaw, Chris. "Detecting Attacks on Networks." *Computer 30,* 12 (December 1997): 16-17.

[S5] Lunt, Teresa F. (SRI International). "A Survey of Intrusion Detection Techniques." *Computers & Security 12,* 4 (1993): 405-418.

[S6] Mukherjee, Biswanath; Heberlein, L.Todd; & Levitt, Karl N. (University of California, Davis). "Network Intrusion Detection." *IEEE Network 8,* 3 (May/June 1994): 26-41 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/papers.html>.

[S7] Denning, Dorothy E. (SRI International). "An Intrusion Detection Model." *IEEE Transactions on Software Engineering SE-13,* 2 (February 1987): 222-232.

[S8] Halme, Lawrence R. & Bauer, R. Kenneth. (Arca Systems). "Ain't Misbehaving—a Taxonomy of Anti-Intrusion Techniques," 163-172. *Proceedings of 18th National Information Systems Security Conference.* Baltimore, MD, Oct. 10-13, 1995. Gaithersburg, MD: National Institute of Standards and Technology, 1995.

[S9] Bishop, Matt; Cheung, Steven; & Wee, Christopher. (University of California, Davis). "The Threat from the Net." *IEEE Spectrum 34,* 8 (August 1997): 56-63 [online]. Available WWW: <URL: http://seclab.cs.ucdavis.edu/papers.html>.

[S10] Mounji, Abdelaziz. (Facultés Univeritaires Notre-Dame de la Paix Namur, Belgium). *Languages and Tools for Rule-Based Distributed Intrusion Detection* [online]. Available FTP: <URL: ftp://ftp.info.fundp.ac.be/pub/users/amo/thesis.ps.Z> (1997).

[S11-1] DARPA. *Darpa ITO Sponsored Research—Challenges* [online]. Available WWW: <URL: http://www.darpa.mil/ito/research/lss/challenges.html> (1988).

[S11-2] Maxion, Roy; Banks, David; & Rocco, Sandy. (Carnegie Mellon University). *Concerning Invictus: Detection of Unanticipated and Anomalous Events—1998 Project Summary* [online]. Available WWW: <URL: http://www.darpa.mil/ito/psum1998/E306-0.html> (1998).

[S11-3] Levitt, Karl. (University of California, Davis). *GlobalGuard: A Protection Architecture for Survivability of Large Scale, High-Confidence Information Networks—1998 Project Summary* [online]. Available WWW: <URL: http://www.darpa.mil/ito/psum1998/F783-0.html> (1998).

[S11-4] Lippmann, Richard & Zissman, Marc. (MIT Lincoln Laboratory). *Intrusion Detection Technology Evaluation—1998 Project Summary* [online]. Available WWW: <URL: http://www.darpa.mil/ito/psum1998/F791-0.html> (1998).

[S11-5] Spafford, Eugene & Winchester, Amber. (Purdue University). *Enhanced Intrusion and Misuse Detection Technology—1998 Project Summary* [online]. Available WWW: <URL: http://www.darpa.mil/ito/psum1998/D848-0.html> (1998).

[S11-6] Tyson, W. Mabry & Linne, Donna. (SRI International). *Explaining and Recovering from Computer Break-ins—1998 Project Summary* [online]. Available WWW: <URL: http://www.darpa.mil/ito/psum1998/E293-0.html> (1998).

[S11-7] Porras, Phillip; Neumann, Peter; & Linne, Donna. (SRI International). *Analysis and Response for Intrusion Detection in Large Networks—1998 Project Summary* [online]. Available WWW: <URL: http://www.darpa.mil/ito/psum1998/E302-0.html> (1998).

[S11-8] Kemmerer, Richard & Mayo, David. (University of California, Santa Barbara). *NetSTAT: A Model-Based Real-Time Intrusion Detection System for Large Scale Heterogeneous Networks—1998 Project Summary* [online]. Available WWW: <URL: http://www.darpa.mil/ito/psum1998/E252-0.html> (1998).

[S12] Information Assurance Technology Analysis Center. *Information Assurance Tools Report* [online]. Available WWW: <URL: http://www.iatac.dtic.mil/iatools.htm> (1999).

[S13] Marshall, Victor H. (Booz, Allen & Hamilton, Inc.). *Intrusion Detection in Computers* [online]. Available WWW: <URL: http://csrc.nist.gov/secpubs/auditool.txt> (1991).

[S14] Lawrence Livermore National Laboratory, Sandia National Laboratories. *Intrusion Detection and Response* [online]. Available WWW: <URL: http://all.net/journal/btbs.ids.html> (1997).

[S15] Information Assurance Technology Analysis Center. "Information Assurance Tools Database." *IATAC Information Assurance Technology Newsletter 1,3* (Spring, 1998): 4-5.

[S16] Lindqvist, Ulf & Jonsson, Erland. (Chalmers University of Technology, Sweden). "How to Systematically Classify Computer Security Intrusions," 154-163. *Proceedings of the 1997 IEEE Symposium on Security & Privacy.* Oakland, CA, May 4-7, 1997. Los Alamitos, CA: IEEE Computer Society Press, 1997 [online]. Available WWW: <URL: http://www.ce.chalmers.se/staff/ulfl/pubs.html>.

[S17] Debar, Hervé; Dacier, Marc; & Wespi, Andreas. (IBM Zurich). *Towards a Taxonomy of Intrusion-Detection Systems* (RZ 3030). Zurich, Switzerland: IBM Research, June 1998 [online]. Available WWW: <URL: http://www.zurich.ibm.com/pub/sti/Security/extern/gsal/docs/>.

[S18] Neuman, Peter G. & Parker, Donn B. (SRI International). "A Summary of Computer Misuse Techniques," 396-407. *Proceedings of 12th National Computer Security Conference.* Baltimore, MD, Oct. 10-13, 1989. Washington, DC: US Government Printing Office, 1989.

[S19] Howard, John D. (CERT/CC). Ch.6, "A Taxonomy of Computer and Network Attacks." *An Analysis of Security Incidents on the Internet—1989-1995* [online]. Available WWW: <URL: http://www.cert.org/research/JHThesis/Chapter6.html> (1997).

[S20-1] Newman, David; Giorgis, Tadesse; & Yavari-Issalou, Farhad. *Intrusion Detection Systems: Suspicious Finds* [online]. Available WWW: <URL: http://www.data.com/lab_tests/intrusion.html> (1998).

[S20-2] Newman, David; Giorgis, Tadesse; & Yavari-Issalou, Farhad. *Intrusion Detection Systems: Suspicious Finds—II* [online]. Available WWW: <URL: http://www.data.com/lab_tests/intrusion2.html> (1998).

[S20-3] Newman, David; Giorgis, Tadesse; & Yavari-Issalou, Farhad. *Intrusion Detection Systems: Suspicious Finds—III* [online]. Available WWW: <URL: http://www.data.com/lab_tests/intrusion3.html> (1998).

[S20-4] Newman, David; Giorgis, Tadesse; & Yavari-Issalou, Farhad. *Intrusion Detection Systems: Suspicious Finds—IV* [online]. Available WWW: <URL: http://www.data.com/lab_tests/intrusion4.html> (1998).

[S20-5] Newman, David; Giorgis, Tadesse; & Yavari-Issalou, Farhad. *Intrusion Detection Systems: Suspicious Finds—V* [online]. Available WWW: <URL: http://www.data.com/lab_tests/intrusion5.html> (1998).

[S20-6] Newman, David; Giorgis, Tadesse; & Yavari-Issalou, Farhad. *Lab Test Vendor Participants* [online]. Available WWW: <URL: http://www.data.com/lab_tests/intrusion_participants.html> (1998).

[S20-7] Newman, David; Giorgis, Tadesse; & Yavari-Issalou, Farhad. *Test Methodology* [online]. Available WWW: <URL: http://www.data.com/lab_tests/intrusion_method.html> (1998).

[S21] Scambray, Joel; McClure, Stuart; & Broderick, John. (InfoWorld Media Group Inc.). "Network Intrusion-Detection Solutions." *InfoWorld 20,* 18 (May 4, 1998) [online]. Available WWW: <URL: http://www.infoworld.com/cgi-bin/displayArchive.pl?/98/18/intrusa.dat.htm>.

[S22] Kumar, Sandeep. (Purdue University). *Classification and Detection of Computer Intrusions* (Coast TR 95-08). West Lafayette, IN: COAST Laboratory, Purdue University, 1995 [online]. Available WWW: <URL: http://www.cs.purdue.edu/coast/coast-library.html>.

[S23] Lane, Terran. (Purdue University). *Machine Learning Techniques for the Domain of Anomaly Detection for Computer Security* (Coast TR 98-11). West Lafayette, IN: COAST Laboratory, Purdue University, 1998 [online]. Available WWW: <URL: http://www.cs.purdue.edu/coast/coast-library.html>.

[S24-1] *Security Magazine. Security Suites* [online]. Available WWW: <URL: http://194.202.195.4/securecomputing/1999_02/testc/products.html> (1999).

[S24-2] *Security Magazine. Security Suites—II* [online]. Available WWW: <URL: http://194.202.195.4/securecomputing/1999_02/testc/products2.html> (1999).

[S25] Surkan, Michael. (PC Week Labs). *Entrax Lags Server Guards* [online]. Available WWW: <URL: http://www.zdnet.com/pcweek/reviews/0615/15entrax.html> (1998).

[S26] McAuliffe, Noelle, et al. (Trusted Information Systems, Inc.). "Is Your Computer Survey Being Misused?—A Survey of Current Intrusion Detection System Technology," 260-272. *Proceedings of the 6th Annual Computer Security Applications Conference.* Tucson, AZ, Dec. 3-7, 1990. Los Alamitos, CA: IEEE Computer Society Press, 1990.

[S27] Graham, Robert. *FAQ: Network Intrusion Detection Systems* [online]. Available WWW: <URL: http://www.robertgraham.com/pubs/network-intrusion-detection.html> (1999).

[S28] Ranum, Marcus, J. (Network Flight Recorder, Inc.). *A Taxonomy of Internet Attacks* [online]. Available WWW: <URL: http://www.clark.net/pub/mjr/pubs/index.shtml> (1999).

[S29-1] IBM Emergency Response Service and the Joint Research Centre of the EC. *First International Workshop on the Recent Advances in Intrusion Detection (RAID 98)* [online]. Available WWW: <URL: http://www.zurich.ibm.com/pub/Other/RAID/RAID98> (1998).

[S29-2] IBM Emergency Response Service and the Joint Research Centre of the EC. *RAID 98—Program* [online]. Available WWW: <URL: http://www.zurich.ibm.com/pub/dac/www/Prog_RAID98/Program.html> (1998).

[S29-3] IBM Emergency Response Service and the Joint Research Centre of the EC. *RAID 98—List of Accepted Papers* [online]. Available WWW: <URL: http://www.zurich.ibm.com/pub/dac/www/Prog_RAID98/Talks.html> (1998).

[S30-1] Larson, Amy K. (*InformationWeek*). *Global Security Survey: Virus Attack* [online]. Available WWW: <URL: http://informationweek.com/743/security.htm> (1999).

[S30-2] Larson, Amy K. (*InformationWeek*). *Global Security Survey: Virus Attack—II* [online]. Available WWW: <URL: http://informationweek.com/743/security2.htm> (1999).

[S30-3] Larson, Amy K. (*InformationWeek*). *Global Security Survey: Virus Attack—III* [online]. Available WWW: <URL: http://informationweek.com/743/security3.htm> (1999).

[S30-4] Larson, Amy K. (*InformationWeek*). *Global Security Survey: Virus Attack—IV* [online]. Available WWW: <URL: http://informationweek.com/743/security4.htm> (1999).

[S30-5] Larson, Amy K. (*InformationWeek*). *Global Security Survey: Virus Attack—V* [online]. Available WWW: <URL: http://informationweek.com/743/security5.htm> (1999).

[S30-6] Larson, Amy K. (*InformationWeek*). *Global Security Survey: Virus Attack—VI* [online]. Available WWW: <URL: http://informationweek.com/743/security6.htm> (1999).

[S30-7] *InformationWeek. Extra Research from the Security Survey* [online]. Available WWW: <URL: http://informationweek.com/743/secure.htm> (1999).

[S30-8] Weston, Rusty. (*InformationWeek*). *Security Survey Methodology* [online]. Available WWW: <URL: http://informationweek.com/743/securit2.htm> (1999).

[S30-9] Larson, Amy K. (*InformationWeek*). *Worldwide Security Priorities* [online]. Available WWW: <URL: http://informationweek.com/743/securit3.htm> (1999).

[S31] Cohen, Fred, et al. (Sandia National Laboratories). *A Preliminary Classification Scheme for Information System Threats, Attacks, and Defenses; A Cause and Effect Model; and Some Analysis Based on That Model* [online]. Available WWW: <URL: http://all.net/journal/ntb/cause-and-effect.html> (1998).

[S32] Foote, Steven. *19 Infosecurity Predictions for '99* [online]. Available WWW: <URL: http://www.infosecuritymag.com/nov/cover.htm> (1998).

[S33-1] Briney, Andy. (*Info Security Magazine*). *Got Security?* [online] Available WWW: <URL: http://www.infosecuritymag.com/july99/cover.htm> (1999).

[S33-2] Briney, Andy. (*Info Security Magazine*). *Under Attack & Underprepared* [online] Available WWW: <URL: http://www.infosecuritymag.com/july99/under.htm> (1999).

[S33-3] Briney, Andy. (*Info Security Magazine*). *Security Overview & Executive Summary* [online] Available WWW: <URL: http://www.infosecuritymag.com/july99/chart1.htm> (1999).

[S33-4] Briney, Andy. (*Info Security Magazine*). *Budgets & Product Purchasing Trends* [online] Available WWW: <URL: http://www.infosecuritymag.com/july99/chart2.htm> (1999).

[S33-5] Briney, Andy & Rose, Barbara. *Study Confirms Increased Security Risks of E-Commerce* [online] Available WWW: <URL: http://www.icsa.net/news/press_room/1999/mag_survey.shtml> (1999).

[S34] *PCWeek Online. PC Week Labs Scoring Methodology* [online]. Available WWW: <URL: http://www.zdnet.com/pcweek/reviews/meth.html> (1999).

[S35-1] *Secure Computing. June 99 Intrusion Detection Market Survey* [online]. Available WWW: <URL: http://194.202.195.4/securecomputing/1999_06/survey/survey.html> (1999).

[S35-2] *Secure Computing. June 99 Intrusion Detection Market Survey—II* [online]. Available WWW: <URL: http://194.202.195.4/securecomputing/1999_06/survey/products_01.html> (1999).

[S35-3] *Secure Computing. June 99 Intrusion Detection Market Survey—III* [online]. Available WWW: <URL: http://194.202.195.4/securecomputing/1999_06/survey/products_02.html> (1999).

[S36] SANS Institute. *The 7 Top Management Errors that Lead to Computer Security Vulnerabilities* [online]. Available WWW: <URL: http://www.sans.org/newlook/resources/errors.htm> (1999).

[S37] Phillips, Ken. (*PC Week*). *One if by Net, Two if by OS* [online]. Available WWW: <URL: http://www.zdnet.com/products/stories/reviews/0,4161,389071,00.html> (1999).

Appendix C: Acronyms

AAFID	An Architecture For Intrusion Detection (using autonomous agents)
AFCERT	Air Force CERT
AFRL	Air Force Research Laboratory
AFIWC	Air Force Information Warfare Center
ANSA	Adaptive Network Security Alliance (ISS)
API	application program interface
ASD	attack signature detection
ASIM	Automated Security Incident Measurement
CCI	Common Content Inspection (Checkpoint)
CIDDS	Common Intrusion Detection Director System
CERT/CC	CERT Coordination Center at Carnegie Mellon University
CIAC	Computer Incident Advisory Capability
CIDF	Common Intrusion Detection Framework
CIO	Chief Information Officer
CISL	Common Intrusion Specification Language
CLIPS	C Language Integrated Production System
CMDS	Computer Misuse Detection System

CMU	Carnegie Mellon University
CORBA	Common Object Request Broker Architecture
COTS	Commercial off the shelf
CSI	Computer Security Institute
CSIRT	Computer Security Incident Response Team
CVE	Common Vulnerabilities and Exposures, also known as Common Vulnerability Enumeration
DARPA	Defense Advanced Research Projects Agency
DMZ	demilitarized zone
DNS	domain name system
EMERALD	Event Monitoring Enabling Responses to Anomalous Live Disturbances
FBI	Federal Bureau of Investigation
FDDI	Fiber Distributed Data Interface
FTP	file transfer protocol
GAO	General Accounting Office
GOTS	Government off the shelf
HTTP	hyper-text transfer protocol
ICSA	International Computer Security Association
ID	intrusion detection
IDES	Intrusion Detection Expert System
IDS	intrusion detection system

IDT	Intrusion Detection Tool
IDSC	Intrusion Detection Systems Consortium (ICSA)
IDWG	Intrusion Detection Working Group (IETF)
IETF	Internet Engineering Task Force
IS	Information system
ISP	Internet service provider
ISS	Internet Security Systems
JAM	Java Agents for Meta-Learning
NASA	National Air and Space Administration
NATO	North Atlantic Treaty Organization
NFR	Network Flight Recorder
NIDES	Network Intrusion Detection Expert System
NSA	National Security Agency
NSTAT	Network STAT
OPSEC	Open Platform for Secure Enterprise Connectivity (Checkpoint)
PI	principal investigator
RAID	Recent Advances in Intrusion Detection (conference)
SAIC	Science Applications International Corporation
SANS	System Administrator and Network Security organization
SEI	Software Engineering Institute
STAT	State Transition Analysis Technology

TCP	transmission control protocol
USTAT	UNIX STAT
VPN	virtual private network

Appendix D: Review of Selected IDS Literature

This appendix is a resource that allows interested readers quick access to documents relevant to their needs. Table D-1 in this appendix provides a road map to the subsequent literature review. Reviews are grouped together according to main and sub-topics. Many of the materials we reviewed are available only on the Web so there is no guarantee that their URLs are still valid; however, we are maintaining a hard copy of all the material reviewed from Web references mentioned in this appendix.

Main Topic	Sub-topic	Subject Matter
1. Surveys		Provides an overview of the status of ID technology.
	1.1 General	Identifies and summarizes techniques and approaches used in ID systems.
	1.2 Tools	Identifies and summarizes characteristics of specific ID tools.
2. Taxonomies		Provides a means for synthesizing knowledge about the subject of intrusion.
	2.1 Intrusion types	Defines a structured framework for comparing and reasoning about different approaches to intrusion.
	2.2 Intrusion methods	Defines a structured framework for comparing and reasoning about different methods for detecting signs of intrusion.
	2.3 Glossaries	Provides a definition of terms relevant to ID.
3. Testing and evaluation		Provides a resource area for those who wish to perform evaluations on ID systems.
	3.1 Product reviews	Provides examples of recent IDS reviews.
	3.2 Vendor questions	Identifies questions to ask before purchasing an IDS.
	3.3 Testing methodologies	Identifies resources (techniques and data) to support a test program.
	3.4 IDS vulnerabilities	Describes weaknesses that ID systems have to being compromised.
4. Research		Provides insights into the evolution of ID technology, and where is it headed.
	4.1 Methods	Reviews significant conceptual approaches to ID.
	4.2 Products	Reviews tools that have pioneered new ID techniques.
5. Commercial products		Reviews primarily commercial literature on current ID systems and their capabilities.
6. ID directions		Discusses issues with current systems and where the research and commercial communities are focussing their efforts.

TABLE D-1: SUMMARY OF LITERATURE REVIEW

D1 Surveys

General surveys

The following papers review general information in intrusion detection. They focus on underlying methods, problems with ID systems, pros and cons of different approaches, adoption issues, etc. They may discuss specific tools but that is not their main focus.

Topic	General ID survey - paper 1
Title	First International Workshop on the Recent Advances in Intrusion Detection
Author(s)/date	Multiple, 1998
Affiliation	Sponsored by IBM, and Joint Research Center of the European Community
Reference	http://www.zurich.ibm.com/pub/Other/RAID/RAID98
Discussion	This workshop brought together leading academic, government and industry players in the ID arena. Presentations covered a comprehensive set of topics associated with intrusion detection including technology advancements, experiences, legal matters, tool development etc. The above reference points to a Web page that summarizes all the presentations.

Topic	General ID survey - paper 2
Title	A Survey of Intrusion Detection Techniques
Author(s)/data	Teresa F. Lunt, SRI International, 1993
Affiliation	SRI International
Reference	Computers and Security, 12 pp 405-418
Discussion	Lunt describes some of the techniques that SRI was exploring in the early 1990s and how these techniques were being tested in the IDES system. A major focus of these explorations was the statistical detection of computer usage patterns that do not correspond to normal user behavior (i.e., anomalies that may indicate intrusion). She also describes some of the early work on the use of expert systems to detect patterns of misuse, for example, password guessing. It was recognized, however, that the expert system "will be no better than the knowledge and the reasoning principles it incorporates," and that "an obvious limitation is that we are looking for known vulnerabilities." In addition to describing the use of statistical and expert system approaches, Lunt reviews the use of neural networks, model-based reasoning, and key-stroke dynamics as techniques being researched to support the detection of unauthorized computer use.

Topic	General ID survey - paper 3
Title	An Introduction to Intrusion Detection
Author(s)/date	Aurobindo Sundaram, 1996
Affiliation	Purdue University
Reference	http://www1.acm.org/crossroads/xrds2-4/intrus.html
Discussion	This introduction to ID covers the basic issues, describing the need and the major approaches that have been used or are being researched. In this way Sundaram reviews anomaly and misuse detection and methods. Under the anomaly category, he reviews training the system through statistical approaches, predictive patterns, and neural networks. Under the misuse category he identifies: rule matching through expert systems; keystroke monitoring, where particular key-strokes combinations may be indicative of a intrusion attempt; model-based ID; and state transition analysis, where the temporal sequence of an attempted intrusion is taken into account. He ends by stating that "intrusion detection is still a fledgling field of research."

Topic	General ID survey - paper 4
Title	Languages and Tools for Rule-Based Distributed Intrusion Detection
Author/date	Abdelaziz Mounji, 1997
Affiliation	Facultes Universitaires Notre-Dame de la Paix Namur
Reference	ftp://ftp.info.fundp.ac.be/pub/users/amo/thesis.ps.Z
Discussion	This dissertation describes a rule-based language called RUSSEL. However, Chapter 2 provides a good review of related work in intrusion detection. Mounji discusses, and provides examples of, tools that employ either the anomaly or misuse approach, and methods that support these approaches. Thus he identifies neural networks, predictive pattern recognition and data clustering as examples of techniques supporting anomaly detection. Within the misuse category he identifies rule base expert systems, state transition, and colored Petri nets as possible approaches. He also identifies benefits and drawbacks of the anomaly approach. Within the misuse category, he only discusses benefits and drawbacks of STAT, a tool that uses a state-transition approach to misuse detection. He concludes the section with a discussion of the current problems of ID systems.

Topic	General ID survey - paper 5
Title	Network- vs. Host-based Intrusion Detection
Author/date	N/A, 1998
Affiliation	Internet Security System

Reference	http://www.iss.net/prod/whitepapers/
Discussion	This short paper is not a survey as the above papers are, but instead "surveys" the characteristics of host and network systems. It provides useful information into the strengths of each approach. Since this is a vendor paper it is not surprising that weaknesses are not identified, although one might imply that the strengths of one approach could be the weaknesses of the other.

Topic	General ID survey - paper 6
Title	An Introduction to Intrusion Detection and Assessment
Author	Rebecca Base, 1999
Affiliation	Infidel, Inc.
Reference	http://www.iss.net/prod/whitepapers/
Content	This very readable report provides current (circa 1999) information on a broad range of issues on intrusion and "explains how ID and vulnerability assessment products fit into the framework of security products." Thus it provides context as to where ID systems fit within the broader scope of computer security, and what ID systems can and cannot do. It also surveys pros and cons on different choices such as selection of a host or network-based system, batch or real time system and signature (anomaly) or statistical (misuse) analysis method.The paper concludes with a useful glossary of terms, but as it is primarily intended as a high-level introduction, does not go into much technical detail.

Tool surveys

While the underlying approaches to ID have some stability, the implementations are rapidly evolving and vendors' products are in constant flux. Thus it is difficult to provide information that does not quickly become obsolete. With that caveat, the following resources provide information on ID tools. We begin by identifying these resources that are primarily lists. These lists tend to mix commercial and research products.

Topic	General IDS resource lists
Title	Intrusion Detection
Author/Date	J. Green, 1998
Affiliation	Information Assurance Technology Analysis Center (IATAC)
Reference	www.iatac.dtic.mil

Title	Michael Sobirey's Intrusion Detection Systems page
Author/Date	M. Sobirey, 6-6-99 (current update)
Affiliation	Security Networks AG
Reference	http://www-rnks.informatik.tu-cottbus.de/~sobirey/ids.html
Title	SANS/NSA Intrusion Detection Tools Inventory
Author/Date	G. Stocksdale, Spring 1998 (no longer updated)
Affiliation	NSA Information Systems Security Organization
Reference	http://www.sans.org/NSA/idtools.htm
Discussion	These resources provide pointers to ID tools. The IATAC list is available only in paper but can be ordered through the Web site. However, there are restrictions on who can order the report (see Web site for details). A summary of this list can be found in the *IATAC Newsletter* Vol 1, No. 3, (also see Web site for details).

Topic	DARPA research
Title	Survivablility of Large Scale Systems
Author(s)	Many, as multiple projects are reviewed
Affiliation	DARPA
References	http://www.darpa.mil/ito/research/lss/challenges.html http://www.darpa.mil/ito/research/lss/projects.html http://www.dyncorp-is.com/darpa/meetings/id98dec/agenda.html http://www.dyncorp-is.com/darpa/meetings/id98feb/agenda.html
Content	These Web pages provide information on what Intrusion Detection Systems DARPA is funding, and in turn provide pointers both to summaries of the work and to the responsible organizations. The latter two URLs point to agendas of DARPA meetings where presentations were given on tools and other activities related to DARPA-funded projects in ID. These agendas point to relevant IDS representations.

Topic	Summary of research and commercial tools
Title	Network Intrusion detection
Author(s)/date	B. Mukherjee, L. T. Heberlein, and K. N. Levit, 1994
Affiliation	UC Davis
References	IEEE Network May/June 1994 http://seclab.cs.ucdavis.edu/papers.html

Discussion	This paper provides a useful summary of many research and commercial tools. The paper is organized into host-based and network-based systems. Each tool review is structured into an overview, a system organization description and a system operation description. The host-based systems covered include Computer Watch, Discovery, HAYSTACK, IDES, ISOA, MIDAS, and Wisdom & Sense. Networked systems include IDES, NADIR, as well as more detailed descriptions of NSM and DIDS. The paper concludes by providing a case study of the algorithmic approach used in HAYSTACK.

D2 Taxonomies

Intrusion detection is still an immature discipline and has yet to establish a commonly accepted semantic framework. Several different classifications of intrusions types have been proposed, as have different ways of classifying ID methods. A commonly accepted vocabulary is still absent. This section identifies papers that address these issues: classification of intrusion types, classification of ID methods, and glossaries of computer intrusion terms.

Taxonomies of intrusion attacks

Topic	A taxonomy of intrusion approaches - paper 1
Title	A Summary of Computer Misuse Techniques
Author(s)	P. G. Neumann, D. B. Parker, 1989
Affiliation	SRI International
Reference	NIST/NCSC 12th Annual Computer Security Conference

Discussion	This paper describes a classification of approaches to computer intrusion. The approach was developed on the basis of 3000 cases that were collected since 1970, and so has a solid practical foundation. The paper identifies nine basic misuse techniques. These span issues ranging from the non-technical (e.g., physical scavenging and spying); to bypassing intended controls (e.g., password guessing and exploitation of incomplete error handling); to preparatory explorations in anticipation of intrusion (such as seeking matches to encrypted password files). The paper raises the issues of collaborative misuse (where multiple individuals, each having different user privileges, are required for perpetrating misuse); effects of misuse (such as compromise of national security or even death); and motivations for misuse (such as espionage or peer pressure or financial gain). Other issues such as skills required for perpetrating misuse, resources required, and avoidance, prevention, detection and recovery are briefly reviewed.

Topic	A taxonomy of intrusion approaches - paper 2
Title	How to Systematically Classify Computer Security Intrusions
Author(s)/date	U. Lindqvist, E. Johnnson, 1996
Affiliation	Chalmers University of Technology
Reference	http://www.ce.chalmers.se/staff/ulfl/pubs.html
Content	Lindqvist and Johnnson's paper describes a set of experiments–the objective "was to find operational measures of computer security, that is measures which reflect the dependence on and uncertainty of the operational environment in a probabilistic way." As a basis for their experiments they adapted the above taxonomy of Neumann and Parker. At the same time they developed a taxonomy of intrusion results based on the goals of computer security: confidentiality, integrity and availability. The experiments used 24 student "attackers" to attempt to creatively penetrate university network (with some restrictions on what they could do). They then correlated the frequencies of intrusion techniques with intrusion results. A conclusion from the paper is that "Some techniques have a one-to-one correspondence with the result, while other techniques can be used to reach many different kinds of results."

Topic	A taxonomy of intrusion approaches - paper 3
Title	An Analysis of Security Incidents on the Internet 1989-1995, Chapter 6: A Taxonomy of Computer and Network attacks

Author/date	J. D. Howard, 1997
Affiliation	CERT(R)/CC
Reference	http://www.cert.org/research/JHThesis/Chapter6.html
Discussion	While this thesis' main goal is to analyze internet incidents, it also provides a framework which supports this analysis. Howard reviews different types of taxonomies including lists of terms; lists of attack categories; lists of results categories; empirical lists (such as Neumann and Parker taxonomy above); two-dimensional correlations (such as Lindqvist and Johnnson's taxonomy, above); and process-based taxonomies which focus on temporal patterns. Howard proposes a taxonomy that fits within the latter category. This shows the sequential dependency between attackers, their tools, the access resulting from the use of these tools, the results generated from the access, and the attackers' objectives in using the results. Each category (e.g. tools) is subdivided into examples (e.g., toolkit).

Topic	A taxonomy of intrusion approaches
Title	A Taxonomy of Internet Attacks
Author/date	M. J. Ranum, no date
Affiliation	Network Flight Recorder, Inc.
Reference	http://www.clark.net/pub/mjr/pubs/index.shtml
Discussion	A set of presentation slides that reviews and categorizes many of the methods used to attack computer networks. Quoting from the presentation, the following types of attacks are reviewed: - social engineering (fooling the victim for fun and profit), - impersonation (stealing access rights of authorized users), - exploits (exploiting a hole in software or operation systems), - transitive trust (exploiting host-host or network-network trust), - infrastructure (taking advantage of protocol or infrastructure features of bugs), - denial of service (preventing the system from being used), and - magic (new things nobody as seen yet)

Taxonomies of intrusion detection methods

Topic	A taxonomy of ID methods - paper 1
Title	Towards a Taxonomy of Intrusion-Detection Systems
Author(s)/date	H. Debar, M. Dacier, A. Wespi, 1998
Affiliation	IBM

Reference	http://www.zurich.ibm.com/pub/sti/Security/extern/gsal/docs/
Discussion	This taxonomy defines four major characteristics of ID systems, and each characteristic is given two sub-categories. These four characteristics are: detection method (either behavior based or knowledge based); behavior on detection (either passive or active); audit source location (either host log files or network packets); and usage frequency (either continuous monitoring or periodic analysis). Each category and sub-category is described and examples are given. Problems, pros and cons are identified where appropriate. At the end of the paper, summary characteristics of 22 tools, with respect to their algorithmic approach and network/host capability, are provided.

Topic	A taxonomy of ID methods - paper 2
Title	Ain't Misbehaving — a Taxonomy of Anti-Intrusion Techniques
Author(s)	L. R. Halme, R. K. Bauer, 1995
Affiliation	Arca Systems, Inc.
Reference	Proc 18th National Information Systems Security Conference, 1995
Discussion	This taxonomy provides a broad perspective on ID-related issues. The authors call the taxonomy AINT (Anti-intrusion Taxonomy) which is composed at the top level of six "mutually supportive" approaches. These are: prevention (precluding the likelihood of attack); preemption (striking offensively against an attacker); deterrence (inhibiting the initiation or continuation of an attack); deflection (deluding the intruder into believing he has succeeded); detection (identifying unauthorized from authorized use); and countermeasures (automatically countering an intrusion). The paper describes each of these categories.

Glossaries of computer intrusion terms

There are a number of glossaries on computer security but few that focus on intrusion detection. The criterion for inclusion in this category was that the glossary had to at least define the terms "anomaly" and "misuse." Sadly, only one glossary was found that met that criterion.

Topic	ID glossary
Title	SANS/NSA Glossary of Terms Used in Security and Intrusion Detection
Author/date	G. Stocksdale, 1998
Affiliation	NSA Information Systems Security Organization

Reference	http://www.sans.org/NSA/glossary.htm
Discussion	Provides definitions for over 200 terms used in computer security and intrusion detection

D3 Testing and Evaluation

Four sub-categories are identified under Testing and Evaluation: product reviews, questions to ask vendors, testing methodologies, and ID systems vulnerabilities. (The latter category is included because it provide insights regarding what to test for.)

Product reviews

Trade magazines perform evaluations of variable quality — some perform testing at reasonable depth while others only do qualitative surveys. By their nature, their evaluations become dated quite quickly due to the rapidly evolving technology. That being said, here are some references to representative evaluations (in chronological order):

Magazine	Review date	Products reviewed	Reference
Secure Computing	February 1999	AXENT eSafe everLink Entrax Notification SAFEsuite SecureWare Snow Vasco	http://194.202.195.4/securecomputing/1999_02/testc/products.html
ZDNet	February 1999	RealSecure SessionWall Kane Netprowler	http://www.zdnet.com/products/stories/reviews/0,4161,389071,00.html
DataComm	August 1998	SessionWall NFR NetRanger NFR RealSecure IDTrack	http://www.data.com/lab_tests/intrusion.html
PCWeek Online	June 1998	Entrax	http://www.zdnet.com/pcweek/reviews/0615/15entrax.html

TABLE D-2: MAGAZINE REFERENCES

Magazine	Review date	Products reviewed	Reference
InfoWorld	May 1998	Abirnet SesionWall IBM solutions RealSecure NFR	Volume 20, Issue 18

TABLE D-2: MAGAZINE REFERENCES

Note that the *DataComm* review is the only one to describe a suite of ID tests that were performed. This might be useful to those wishing to perform their own evaluations. The *PCWeek* approach to the evaluation of software products (not just ID systems) can be found at http://www.zdnet.com/pcweek/reviews/meth.html.

Comments from these reviews indicated that while they perform a useful function, current-day ID systems are still immature. The *DataComm* reviewer states: "Sure, ID systems spot attacks as advertised—on empty networks. They also work well on heavily utilized Ethernet segments. But fill up a fast Ethernet segment with traffic and that vigilance vanishes; in fact, no product detected all the attacks when the network was heavily loaded." The *InfoWorld* reviewer commented, "By the end of our testing we were somewhat underwhelmed by the current state of the technology and its usefulness. In essence we see these solutions as little more than sophisticated packet analyzers."

Selection criteria

A few papers provide useful information on what to ask prospective vendors. In particular the Computer Security Institute has two informative articles in this area.

Topic	Questions to ask vendors
Title	Tough Questions for IDS Vendors
Authors/date	C. Klaus (ISS), G. Spafford (COAST), L.Sutterfield of Cisco, M. Ranum (NFR), 1998
Affiliation	Computer Security Institute
Reference	http://www.gocsi.com/intrusion.htm
Title	CSI Intrusion Detection System Resource
Author/date	R. Power and R. Farrow, 1998
Affiliation	Computer Security Institute
Reference	http://www.gocsi.com/ques.htm
Title	CSI asks the tough questions
Author/date	N/A, 1998
Affiliation	Centrax Corporation
Reference	http://www.centraxcorp.com/magazine.html

Title	FAQ: Network Intrusion detection Systems
Author/date	Robert Graham, 1999
Affiliation	N/A
Reference	http://www.robertgraham.com/pubs/network-intrusion-detection.html
Title	A Selection Criteria for Intrusion Detection Systems
Author/date	E. Amoroso, R. Kwapniewski, 1998
Affiliation	AT&T Bell Labs
Reference	Proceedings of the 14th Annual Computer Security Applications Conference, Phoenix, Arizona, 1998
Discussion	The Centrax Corporation responds to the "tough" questions identified in the Computer Security Institute paper. The paper *FAQ: Network Intrusion Detection Systems* discusses some of these questions, providing some further insight. The latter paper covers a broad range of issues with ID systems. The last paper provides a set of criteria (structured into detection, response and deployment categories) for comparing and assessing ID systems. It contains an appendix that provides a vendor questionnaire.

Testing intrusion detection systems

One reflection of the lack of maturity in current ID systems is the lack of testing suites or methodologies. This section provides pointers to documents that have addressed these issues.

Topic	IDS testing at Lincoln Labs
Title	DARPA Intrusion Detection Evaluation
Author/date	R. Lipmann, M. Zissman, 1999
Affiliation	MIT Lincoln Labs
Reference	http://www.ll.mit.edu/IST/ideval/index.html
Title	1998 Project Summary - Intrusion Detection Technology Evaluation
Author/date	R. Lipmann, M. Zissman, 1998
Affiliation	MIT Lincoln Labs
Reference	http://www.darpa.mil/ito/psum1998/F791-0.html
Title	Intrusion Detection System Evaluation
Author/date	R. Lipmann, M. Zissman, 1998
Affiliation	MIT Lincoln Labs
Reference	Information Assurance Newsletter Vol. 2, No. 2 (from the DoD sponsored Information Assurance Technology Analysis Center)

Discussion	These documents describe the work at Lincoln Labs for "collecting and distributing the first standard corpus for evaluation of computer network ID systems." These data will focus on measuring the "probability of detection and probability of false-alarm for each system under test."

Topic	IDS testing at UC Davis
Title	A Methodology for Testing Intrusion Detection Systems
Author/date	N. Puketza, K Zhang, M. Chung, B. Mukherjee, R. Olsson, 1995
Affiliation	University of California, Davis
Reference	http://seclab.cs.ucdavis.edu/papers.html
Title	Simulating Concurrent Intrusions for Testing Intrusion Detection Systems: Parallel Intrusions
Author/date	N. Puketza, K Zhang, M. Chung, B. Mukherjee, R. Olsson, 1997
Affiliation	University of California, Davis
Reference	http://seclab.cs.ucdavis.edu/papers.html
Title	A Software Platform for Testing Intrusion Detection Systems
Author/date	N. Puketza, K Zhang, M. Chung, B. Mukherjee, R. Olsson, 1997
Affiliation	University of California, Davis
Reference	http://seclab.cs.ucdavis.edu/papers.html This paper can also be found in IEEE Software, Sept/Oct, 1997
Discussion	These papers describe an IDS testing environment that simulates network traffic, both normal and malicious. It allows one to create scripts that can be recorded and played back in order to test different target ID systems with the same data. Two approaches to testing are allowed–sequential and concurrent. The former simulates one attacker while the latter simulates a cooperative attack originating form multiple locations. There are three categories of testing procedure: intrusion identification (focusing on the IDS ability to distinguish intrusions for normal behavior); resource usage (for evaluating the system resources used); and stress testing (to assess the ID System's ability to detect misuse under high load). The Network Security Monitor (NSM) tool was used to evaluate its capability by using scripts that simulated several attacks such as password cracking. However, the documentation focuses more on the capability of NSM and intrusion detection than on the test environment, and provides few insights into what was learned about constructing such environments.

Topic	IDS testing at IBM
Title	An Experimental Workbench for Intrusion Detection

Authors/date	H. Debar, M. Dacier, A. Wespi, S. Lampart., 1997
Affiliation	IBM, Zurich
Reference	http://www.zurich.ibm.com/pub/sti/Security/extern/gsal/docs/
Discussion	This paper focuses on the design principles and implementation of a an experimental workbench for comparative evaluation of ID systems. "This workbench enables us to compare the respective efficiency of our prototypes in terms of, for example, false alarm rates."

Topic	IDS testing at the U.S. Air Force
Title	Testing and Evaluating Computer Intrusion Detection Systems
Authors/date	R. Durst, T Champion, B. Witten, E. Miller, L. Spagnuolo, 1999
Affiliation	SenCom Corp. and Air Force Research Laboratory
Reference	Communications of the ACM, July, 1999
Discussion	The paper describes the development of an IDS testbench and the resulting testing of three DARPA-funded ID systems (plus a GOTS IDS), and summarizes quantitative results. In preliminary conclusions it indicated, among other things, that signature-based detection was effective in reducing false alarm rates, but that string matching as implemented in most network-based systems has high false alarm rates and misses most kinds of attack. In conclusion the paper stated that "Recent major acquisitions totalling hundreds of millions of dollars for little more than ad-hoc security solutions indicate a desperate, indiscriminate need for computer security. If it sounds like panic, it may very well be."

IDS Vulnerabilities

Intrusion detection systems are themselves vulnerable to attack. Since knowledge of weaknesses is a prerequisite to knowing how to test ID systems, this section provides some relevant resource materials. In this section we identify four papers. These are listed in the following table.

Topic	IDS Vulnerabilities
Title	Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection
Author/date	T. Ptacek, T. Newsham, 1998
Affiliation	Network Associates
Reference	http://www.clark.net/pub/roesch/public_html/IDSpaper.pdf
Title	50 Ways to Defeat your Intrusion Detection System
Author/date	F. Cohen, 1997
Affiliation	Fred Cohen & Associates

Reference	http://www.all.net/journal/netsec/9712.html
Title	Defeating Sniffers and Intrusion Detection Systems
Author/date	Horizon, 1998
Affiliation	Phrack Magazine
Reference	http://pulhas.org/phrack/54/P54-10.html
Discussion	The first paper claims that "there is insufficient information available in packets read off the wire to correctly reconstruct what is occurring inside complex protocol transactions, and next, that ID systems are inherently vulnerable to denial of service attacks. The first of these problems reduces the accuracy of the system, and the second jeopardizes its availability." To back up their case, the authors describe a set of tests that were performed on four popular commercial products. The report indicates that "Every IDS we examined could be completely eluded by a savvy attacker." This detailed paper concludes with some concerns about the current state of testing: • No credible public evaluation of network ID systems currently exist. The trade press evaluates security products by their features and ease of use, not their security. • One issue that drastically impacted our ability to test ID systems was the availability of source code. • If this work makes anything clear, it's that marketing claims cannot be a trusted source of information about ID systems. This second paper takes a semi-serious look (with serious implications) at the multitude of ways it is possible to elude an IDS. In the last paper a hacker describes how to defeat ID systems — and provides the code to do it.

D4 Research

The research category is divided into "methods" and "tools." This is a somewhat artificial distinction since much research is done through exploration with tools. However, the "methods" category covers research that does not result in any major research products (although some software may be developed). The research papers are, by definition, forward looking, but differ from the category below "IDS directions" by being more technically and

exploration oriented. The "IDS directions" category addresses broader, more qualitative issues.

Methods

The "methods" category summarizes advanced work performed in the areas of neural networks, genetic algorithms, inductive rule generation, pattern recognition, and data fusion. There is a significant machine learning flavor to these more conceptual approaches, which have in common

- the need to train the IDS using raw data (this will probably be audit data)
- the need to provide the learning component of the IDS with guidance as to what is normal and is abnormal behavior
- the fact that computer-generated decision rules may be opaque to human interpretation

Topic	Neural network applications
Title	A Survey of Intrusion Detection Techniques
Author/date	T. Lunt, 1993
Affiliation	SRI International
Reference	Computers & Security, 12, (1993) pp 405-418
Title	Typing your ID via AI
Author/date	L. Eliot, 1995
Affiliation	Eliot and Associates
Reference	AI Expert
Title	Neural Networks Applied in Intrusion Detection Systems
Author/date	J. Bonifácio Jr., A. Cansian, A. de Carvalho e E. Moreira, 1998
Affiliation	University of Sao Paulo
Reference	Published in the Proceedings of the IEEE World Congress on Computational Intelligence, WCCI'98, Anchorage, USA

Discussion	Neural networks can be trained to recognize patterns in data. It has therefore been suggested that neural networks be trained to recognize dynamic keystroke characteristics as a means of intrusion detection (see first two papers in this table). The approach is based on the assumption that each computer user has typing characteristics that are unique, cannot be easily duplicated by others, and whose signature does not change rapidly with time. Keystroke sequences are captured and analyzed for patterns that identify the individual. Lunt's paper only briefly refers to this application. Eliot's paper discusses the use of a multi-layered neural net to learn the timing patterns between input strokes, as explored by AT&T researchers. The third paper looks at training neural networks to recognize patterns of intrusion. Training was performed using attack patterns on different Internet services. The neural net was then used to detect similar attacks. For new patterns, the network was adoptively retrained. Preliminary results on test data indicated an average misclassification rate of five percent.

Topic	Genetic algorithm applications
Title	Genetic Algorithms, an Alternative Tool for Security Audit Trails Analysis
Author/date	L. Me, 1994
Affiliation	Université de Rennes
Reference	www.supelec-rennes.fr/rennes/si/equipe/lme/these/these-lm.html
Title	Active Defense of a Computer System Using Autonomous Agents
Author/date	M. Crosbie and G. Spafford, 1995
Affiliation	Purdue University
Reference	www.cs.purdue.edu/coast/coast-library.html

Discussion	Inspired by biological evolution, genetic algorithms are based on artificial genes (string structures) that carry messages of varying "fitness" to perform a task. Depending on their fitness (a measure that is predefined), more successful genes preferentially exchange their message sequences with other genes. The resulting new gene combinations are propagated to subsequent generations with resulting improvements to the gene pool. This technique has been investigated as a means to teach a set of artificial genes to recognize patterns of intrusion. Gene learning is performed off-line (i.e., the genes are defined using audit data) and then applied either in real time or in batch mode. The approach does have the advantage that the data drives the understanding of what constitutes intrusive behavior, rather than subjective human experience. However, the field is still immature and being explored; there are no operational tools yet available. See D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.

Topic	Rule induction applications
Title	Learning Patterns from UNIX Process Execution Traces for Intrusion Detection,
Author/date	W. Lee and S. Stalfo, 1997
Affiliation	Columbia University
Reference	AAAI Workshop: AI approaches to fraud detection and risk management, AAAI press
Title	Data Mining Approaches for Intrusion Detection
Author/date	W. Lee and S. Stolfo
Affiliation	Columbia University
References	www.cs.columbia.edu/~sal/hpapers/USENIX/usenix.html
Title	Security Audit Trail Analysis Using Inductively Generated Predictive Rules
Author/date	H Teng, K. Chen, and S. Lu, 1990
Affiliations	Digital Equipment Corp, University of Illinois
Reference	Proceedings of the Sixth Conference on Artificial Intelligence Applications

Discussion	Several research groups have investigated rule induction as a means to identifying intrusions. As with genetic algorithms, the audit data is used to train a system as to what constitutes intrusive behavior. However, the inductive approach generates a set of if-then rules from examples of data sequences that represent normal and abnormal behavior. (See the discussion on the JAM system below.) Pattern recognition comprises another approach to intrusion detection through machine learning. Much of this work has been performed by the COAST Project at Purdue University (http://www.cs.purdue.edu/coast/).

Topic	Pattern matching applications
Title	An Application of Pattern Matching in Intrusion Detection
Author/date	S. Kumar, E. Spafford, 1994
Affiliation	Purdue University
Reference	www.cs.purdue.edu/coast/coast-library.html
Title	Matching Learning Techniques for the Domain of Anomaly Detection for Computer Security
Author/date	T. Lane, 1998
Affiliations	Purdue University
Reference	www.cs.purdue.edu/coast/coast-library.html
Discussion	The first of the two papers focuses on misuse detection, and attempts to address several challenging ID issues. These include the challenge of digesting large quantities of data in real time, and the use of multiple identities in the perpetration a single attack. The approach uses colored Petri nets to identify partially ordered sequences of attack components. It provides a three-layered architecture (the information layer, the signature layer, and the matching engine) that allows for portability and ease of signature management. The second paper focuses on anomaly detection, and is based on a hierarchical model of user behavior. The leaves of the hierarchy reflect the users' temporal input patterns, while higher nodes reflect increasing abstractions of user behavior. Learning occurs from sequences of user input patterns using a hidden Markov model. HMMs allow one to train the IDS to reflect stochastic transitions between the user-induced patterns. In this way IDS learns user behavior that can be stored to assess future behavior. the authors claim that the approach is also able to address issues of conceptual drift, trusted insiders, and hostile training.

Topic	Computer immunology

Title	Computer immunology
Author/date	S. Forrest, S. Hofmeyr, A. Somayja, 1996
Affiliation	University of New Mexico
Reference	http://www.cs.unm.edu/~forrest/papers.html
Title	A Sense of Self for UNIX Processes
Author/date	S. Forest, T Longstaff
Affiliation	University of New Mexico, Software Engineering Institute
Reference	http://www.cs.unm.edu/~forrest/papers.html
Title	Distributed Approach to Anomaly Detection
Author/date	P. D'haeseleer, S Forrest, P. Helman
Affiliation	University of New Mexico
Reference	http://www.cs.unm.edu/~forrest/papers.html

Discussion

Computer immunology is inspired by biological immunology in that systems having a sense of "self" are able to detect and remove foreign objects. In biological systems, such foreign objects include bacteria and viruses. This work explores the application of immunological ideas to computer intrusion detection. In this context, the computer can be taught to detect sequences of instructions that may be hostile to the operation of the system.

The approach is based on the generation of foreign-body detectors that are analogous to antibodies and other immune system components. The referenced papers describe various implementations of this idea. In one implementation of the approach (See *A Sense of Self for UNIX Processes*), these detectors consist of short sequences of system calls that match sequences observed during periods in which the system is known not to be subject to attack. Sequences that do not match a detector are considered suspect.

Preliminary experiments described in the above paper have focussed on UNIX applications such as *sendmail*. Fixed-length sequences of system calls, reflecting normal behavior for the sendmail application, were first collected to generate a database of normal activity. In a second phase, new sequences (reflecting other application such as *ps* and *finger*) were considered "illegal," and were compared with the sequences in the database and found not to match, indicating that each process has a unique sense of "self." A second set of experiments looked at known attack scripts such as *sunsendmailcp*. This generated sequences that were not in the normal database, indicating that abuse results in abnormal system call signatures. Preliminary results indicated that this approach has promise, but much has still to be done. The approach has the advantage of not requiring knowledge about the nature of the intrusion. Normal usage by an intruder will not be detected and as yet there are no mechanisms to let the detector system modify itself in response to changing patterns of non-intrusive behavior.

Available Research Systems

Over the past 15 years, a number of research-oriented ID systems have been developed. Many of these are no longer supported. While a significant number of such systems are identified in references in Appendix B [S6, S12, S26, B3, B4], many of the sources they cite can no longer be accessed. For example, a reference from the Information Assurance Technology Analysis Center [S12] identifies 15 academic systems with URLS, however just 18 months after the publication of the reference, only eight of these URLs are currently valid.

The number of current research projects is relatively small. Some of these have been reviewed in the body of the report. In this section we focus on projects that are active and have software that can be downloaded. Keep in mind that these systems are experimental and that the availability of the software is subject to change.

Topic	Research product - AAFID
Title	An Architecture for Intrusion Detection Using Autonomous Agents
Author(s)/data	J. Balasubramaniyan, J Garcia-Fernandez, D. Isaacoff, E. Spafford, D. Zamboni, 1998
Affiliation	Purdue University
Reference	www.cs.purdue.edu/coast/projects/autonomous-agents.html
Discussion	AAFID is being built to explore architectural issues associated with distributed ID. The system is built from three basic components: autonomous agents (each of these monitors specific aspects of activity on a host and reports abnormal behavior); transceivers (that reside on hosts and are responsible for receiving data from the agents and controlling agent's activities, e.g., starting and stopping); and monitors (that coordinate activities between transceivers). While AAFID is not a network -based system, it does support some network functions. The agent approach was taken because of the flexibility and robustness that such an approach provides. It is flexible in that agents can be inserted, removed, and upgraded without having a significant impact on the real-time operation of the system. It is robust in that, if isolated or local groups of agents are compromised, or somehow fail, then the operation of the wider system is not necessarily jeopardized. Two prototypes have been built to date. The second of these, AAFIDS2, is built in Perl for portability, and is publicly available. It includes a set of existing agents, transceivers and monitors, and provides the support for implementing new agents, transceivers and monitors.

Topic	Research product - Hummer
Title	Trust Mechanisms for Hummingbird
Author(s)/data	J Evans, D. Frincke
Affiliation	University of Idaho
References	www1.acm.org/crossroads/xrds2-4/humming.html www.csds.uidaho.edu/~hummer/home.html

Discussion	Hummer is the product of the Hummingbird project. This project examines the problem of cooperation between computer sites that are facing common intrusion problems, but who may not fully trust each other. Hummer allows such sites to exchange data regarding the intrusions without compromising security or confidentiality. Hummer has three main components: the Message Distribution Unit (that communicates with Hummers at other sites), a Data Distribution Unit (that assesses what data should be sent to which Hummer), and the Data Collection Unit (that performs the actual data collection). The latter unit embeds in it multiple independent tools (such as CMDS, or Shadow) that perform local ID. Thus a Hummer does not perform any actual intrusion analysis itself. Rather it provides a cooperative framework that embeds within it existing ID units, and allows interactions to take place between sites that may differing levels of trust. Currently, a limited Hummer test has been performed. This test actually involved only a single network, but the components on that network were configured to emulate four virtual networks.

Topic	Research product - JAM
Title	JAVA Agents for Meta-learning: Fraud and Intrusion Detection in Financial Information Systems
Author(s)/data	S. Stolfo
Affiliation	Columbia University
Reference	http://www.cs.columbia.edu/~sal/JAM/PROJECT/

Discussion

A high-level objective of the JAM project is similar to the that of the Hummingbird project — both address issues of distrust between communications organizations. However, the JAM project focuses specifically on cooperation between financial institutions that have to address issues of financial fraud, and is less focused on distributed architecture. In the financial context, competitive and legal reasons may prohibit sharing financial data. However, it is argued that these institutions can still cooperate through the sharing of fraudulent transaction models.

JAM (Java Agents for Meta-Learning) is a system that learns fraudulent patterns through training on large numbers of credit card records, some of which reflect illegal transactions. The program provides support for training using a variety of different classifier algorithms. Classification rules are developed at different financial institutions (reflecting local conditions). Because these rules do not constrain sensitive data, they can be shared. Through a meta-learning agent, the rules sets can are integrated, thus combining the intrusion characteristics derived from different sources. This paper claims that this integration improves overall accuracy in detecting fraudulent activity.

JAM has been evaluated against the Lincoln Lab tests (see the review section above on testing). The program is also available for download from the University of Columbia Web site.

Topic	Research product - NID
Title	LLNL's Network Intrusion Detector Site
Author(s)/data	N/A, 1999
Affiliation	Laurence Livermore National Laboratory
Reference	http://ciac.llnl.gov/cstc/nid/nid.html

Discussion

Laurence Livermore's Network NID product is a network-based system that provides protection against intrusive behavior to a secure subdomain of hosts on the network. Like most network-based systems, it resides on its own machine and passively collects data from the network. It can provide ID support in three different ways: by analyzing traffic in real time, by performing analysis off-line, and by gathering statistics for later review. NID can perform signature analysis, looking for suspicious pattern within the packets. It can detect and issue alerts on activity associated with vulnerabilities that may have serious consequences, and it can detect network attacks such as port scans or SYN floods.

NID provides the ability to specify a variety of filter mechanisms on which to alert. First, it allows one to define the host or sub-network addresses that constitute the secure domain and which are to be monitored. Second, one can define the Internet services that need to be monitored. Third, one can define a domain boundary and specify the direction of the traffic across the boundary that must be monitored.

NID is freely available to U. S. Department of Energy facilities and contractors. It is also available to U.S. Government civilian agencies. It runs on various versions of HP-UX, Solaris, SunOS, and Linux.

D5 Commercial products

The field of intrusion detection is expanding so rapidly that it is difficult to keep current with changes to existing products and the introduction of new products. There are three notable sources of information on ID systems. These are Michael Sobirey's ID Systems Page with 80 entries [B3], the SANS/NSA ID tools Inventory with 38 entries [B4], and the Information Assurance Technology Analysis Center's report on Intrusion Detection with 42 entries [S12]. The latter report (in hard copy) is available to U. S. DoD-related organizations and to organizations registered with the Defense Technical Information Center (DTIC). The sources also identify non-commercial products. Please note that, as of this writing, none of these sources is completely current.

In addition to the products covered in Section 4, the following table summarizes some additional commercial products.

Topic	Commercial products
Title	Kane Security Monitor
Product type	Host-based; NT platforms
Affiliation	Security Dynamics
Reference	http://www.securitydynamics.com/products/datasheets/ksmd.html
Title	POLYCENTER
Product type	Host-based; Digital UNIX OS
Affiliation	Compaq
Reference	http://www.digital.com/info/security/id.htm
Title	CyberCop Monitor
Product type	Host/Network-based; NT and UNIX platforms
Affiliation	Network Associates
Reference	http://www.nai.com/products/security/cybercop_scanner/monitor.asp
Title	Network Security Agent
Product type	Network-based; dedicated RISC system
Affiliation	Touch Technologies
Reference	http://www.ttisms.com/tti/nsa_www.html
Title	Node Sentry
Product type	Network-based; HP OpenView platforms
Affiliation	Hewlett Packard
Reference	http://www.openview.hp.com/products/node/features
Title	SecureNet Pro
Product type	Network-based; Solaris, Linux, FreeBSD-x86, BSDi-x86 platforms
Affiliation	MimeStar
Reference	http://www.mimestar.com/html/data_sheet.htm
Title	SessionWall-3
Product type	Network-based, Windows 95 and NT
Affiliation	Platinum Technologies
Reference	http://abirnet.com/products.html
Title	RealSecure
Product type	Network-based; Solaris and NT platforms
Affiliation	Internet Security Systems
Reference	http://www.iss.net/press_rel/whatsnew.php3#RS
Title	T-sight
Product type	Network-based, Windows NT
Affiliation	En Guard Systems
Reference	http://www.enguard.com/software/t-sight/everview.html

Discussion	The ID systems identified above are typical of those available at the time this report was published. The majority are network-based systems; a few are host-based and only one has both network and host-based capability. However, given that both technologies have their strengths and weaknesses (e.g., network-based systems tend to generate many false positives, while host-based systems may be more vulnerable to attack), a combination of both is desirable and we may see more systems combining the technologies. While some of these systems claim to be "anomaly" detectors, in fact true anomaly detection (i.e., the detection of learned behavioral patterns) is not often seen. Note that in the above information boxes, the "Authors/Date" line has been replaced with "Product type."

D6 ID Directions

Intrusion detection systems are still in their infancy. However, improving the technology is like shooting at a moving target — for every advance that is made, additional vulnerabilities are discovered and exploited. The papers identified below examine the state of the technology, identify problems with current systems, and provide some indication of where intrusion detection is headed.

Topic	Security trends
Title	Global Security Survey: Virus Attack
Author(s)/data	A. Larson, 1999
Affiliation	Information Week
Reference	http://www.informationweek.com/743/security.htm

Discussion	While not specifically focused on ID systems, this survey of 2700 security professionals provides some interesting insights relevant to intrusion detection. First, respondents believed that 48% of all breaches and espionage was caused by hackers and terrorists. (This number was only 14% in 1988.) Second, the number of security problems blamed on insiders was down from 58% in 1988 to 41% in 1999. Thus the often-stated observation that most illegal activity originates from inside the organization appears to be changing. By far the largest barrier to not employing effective security is "lack of time"(25%). This is followed by five additional reasons, all of which fall into the 12%-13% response range. These are: complexity of the technology, pace of change, lack of management support, capital expense, and lack of qualified staff. Finally, 39% of organizations indicated that they would be installing ID systems in 1999. While this is half the number intending to install firewalls, it would still appear to be a substantial number.

Topic	Expert perspectives on today and tomorrow
Title	CSI Roundtable: Experts Discuss Present and Future Intrusion Detection Systems
Author(s)/data	R. Power with M. Ranum, C. Klauss, D. Curry, G. Spafford, L. Sutterfield, 1998
Affiliation	Computer Security Institute
Reference	http://www.gocsi.com/roundtable.htm

Discussion	This lively roundtable, conducted by industry and research experts in the field, covered a wide variety of intrusion detection issues. First, unreasonable expectations of future systems were identified. These included the ability to response to unknown attacks, the ability to detect slow attacks, and automatic "hands-off" response to attacks. Current big issues were seen to be the scalability and performance of ID systems, data overload and fine-tuning the system in order to reduce data overload, the need for support of data interpretation and consequent actions, and the number of security holes in existing applications. The panel saw the following issues as being important in the future: better databases of attack signatures; distributed multiple-sensor based ID systems (that can lead to an integrated view of the "big picture" ID systems that are simpler to operate and more modular); integration of "burglar alarm"; and expert systems approaches to intrusion detection and automated response. It was noted that research tends not to address the issue that "real security equals network management." Research systems tend to have non-intuitive interfaces and documentation, and are hard to manage. There was significant disagreement on where to place network ID systems — in front of the firewall or behind. Reasons identified for placing the IDS in front included global visibility on what was happening, and the ability to see early signs of impending attacks or sweeps. Reasons for placing the IDS inside the firewall included: indifference to what's happening beyond the firewall; enhanced prevention of insider abuse; and enhanced detection of the firewall's effectiveness.

Topic	Fusing diverse information sources
Title	Intrusion Detection Systems and Multisensor Data Fusion: Creating Cyberspace Situational Awareness
Author(s)/data	T. Bass, 1999
Affiliation	Silk Road Group Ltd.
Reference	http://www.siklroad.com/paper/html/ids/node1.html
Title	A Glimpse into the Future of ID
Authors/data	T. Bass, D. Gruber,1999
Affiliations	Silk Road Group Ltd., Langley AFB,
Reference	http://www.siklroad.com/paper/html/ids/node1.html http://www.siklroad.com/paper/html/glimple/ (also USENIX Association magazine, July, 1999)

Title	A Cop on the Beat: Collecting and Appraising Intrusion Evidence
Author(s)/data	T. Goan
Affiliation	Stottlet Henke Associates, Inc.
Reference	Communications of ACM, July 1999
Discussion	These papers attempt to address the weaknesses in the current generation of ID systems; namely, their inability to provide accurate diagnosis of intrusions, and their tendency to generate multiple false alarms. In general, these papers call for technologies that allow information from diverse and possibly redundant sources to be fused together, providing a more robust picture of what is happening and giving some level of confidence in the resulting conclusions. The first paper reviews current systems, and identifies why they are not adequate to defend against sophisticated attackers. It is suggested that "the art and science of data fusion is directly applicable in cyberspace for intrusion and attack detection." Use of data fusion is challenging as it requires a multifaceted approach to intrusion detection, both with respect to the diversity of data sources and the diversity of technologies that may be brought to bear in the analysis and integration of that data. Examples of applicable technologies include statistics, artificial intelligence, operations research, pattern recognition and decision theory. The second paper provides additional information from the same author and presents an interesting analogy between aircraft control monitoring and cyberspace monitoring. The final paper uses the analogy of Joe, the "cop on the beat" and his approach to obtaining and analyzing evidence of illegal activity. Such issues as prioritizing effort, committing limited resources, acquiring help, and the rights of others, require consideration. The paper then reviews a research effort instantiated in a tool called ICE (Intelligent Correlation of Evidence). This tool uses multiple techniques to analyze evidence and is supported by "adaptable evidence collection" implemented through Bayesian reasoning in conjunction with machine learning.

Topic	Intrusion detection standards
Title	Intrusion Detection Exchange Format (idwg)
Author(s)/data	M Erlinger, S. Staniford-Chen, N/A
Affiliation	Aerospace Corporation, University of California at Davis
Reference	http://www.ietf.org/html.charters/idwg-charter.html
Title	The Common Intrusion Detection Framework Architecture

Author/data	P. Porras, D. Schnackenberg, S. Staniford-Chen, M. Stillman, F Wu, N/A
Affiliation	SRI, Boeing, UC Davis, Odyssey Research, NCSU
Reference	http://seclab.cs.ucdavis.edu/cidf/
Discussion	Currently the ID field is a little like the wild west as reflected by organizational and technological instability. One symptom of this is the lack of standards on interoperability between tools. There are, however, some attempts being made to correct this. Under the auspices of the Internet Engineering Task Force, the Intrusion Detection Working Group (IDWG) is developing a set of functional requirements and protocols for communication between ID systems, and specifying an ID language that describes data formats. DARPA is pursuing an effort to develop a Common Intrusion Detection Framework (CIDF). Members of this group overlap with the IDWG so that there is some commonality. This effort focuses on establishing a high-level architectural view of IDS functional components, and the data communication needs between these components. The major components are called event generators, event analyzers, event databases, and response units. In a field as immature, rapidly evolving, and highly competitive as the intrusion detection field is, can standards have a significant impact?

Topic	On the maturity of ID technology - paper 1
Title	Security on Internet Time
Author(s)/data	Marcus Ranum, N/A
Affiliation	Network Flight Recorder, Inc.
Reference	www.clark.net/pub/mjr/pubs/index.html

Discussion	The papers in this and the subsequent two tables identify problems in the current generation of ID systems and indicate the challenges for the future. The first paper (a presentation) looks at security technology in general, but raises issues that are particularly germane to ID technology. In particular, the paper is quite skeptical of the influence of standards bodies that are dominated by vendors whose objectivity may be questioned. In addition, the author believes that a large installed user base for a product may result in too much influence on a standard. Among his other concerns are the fact that security software releases are, in effect, "beta" versions that customers are forced to help fix, and that security architectures are too complex. He believes that, while "Internet time" has been a boon to the software industry's growth, it has significantly hurt security. An implication of his remarks is that poor software engineering of systems makes security weak, and this to some extent has resulted in the need for ID systems. While he provides some potential remedies, his prognosis for the future is that "the situation is bad and getting worse."

Topic	On the maturity of ID technology - paper 2
Title	Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection
Author(s)/data	T. Ptacek, T. Newsham, 1008
Affiliation	Network Associates, Inc.
Reference	http://www.clark.net/pub/roesch/public_html/IDSpaper.pdf
Discussion	The second paper, also reviewed under the heading *Testing and Evaluation,* addresses specific weaknesses in the current generation of ID systems. It identifies two new (circa 1998) classes of intrusion that have implication for future IDS functionality. In the first of these attacks, called an "insertion attack," the attacker manipulates a packet's contents so that the IDS accepts the packet (thus disguising the message) that the target host rejects. In the second type of attack, called an "evasion attack," the attacker manipulates a packet's contents so that the IDS rejects the packet (thus also disguising the message) accepted by the target host. Both situations exploit the fact that the IDS may interpret package reassembly in a different manner than the intended host. The paper indicates that very basic problems were uncovered with ID systems, thus implying that little vendor testing to stress the tools had been performed. This is certainly consistent with the observations of the above paper about the premature release of products still in the "beta" phase.

Topic	On the maturity of ID technology - paper 3
Title	Intrusion Detection and Response
Author(s)/data	F. Cohen, 1996
Affiliation	Lawrence Livermore National Laboratory, Sandia National Laboratories
Reference	http://all.net/journal/ntb/ids.html
Discussion	The final paper reviews the state of the practice and art in intrusion detection systems. Cohen states that only modest improvement in IDS technology has occurred over the past decade. In this regard he cites the use of techniques such as artificial intelligence, automated detection to reduce false negatives, and systems that operate across different computing systems. He believes that "the most important innovation has been the combination of audit records from multiple sources and the automated retrieval of irrelevant records." Looking to the future, Cohen sees research focusing in the following areas: basic definitions (see Appendix A of this report) and mathematical understanding, metrics for comparing systems with each other or to a common standard, weaknesses in ID systems that could make them ineffective against skilled attackers, consistency and content from information sources, damage assessment and recovery, and unlimited scalability. He also raises the issues of eliminating false positives and negatives, the challenge of testing ID systems, and assessing damage from an attack as important issues. The article provides an interesting review of ID techniques that are applied to other technologies such as the telephone network, the power grid, and satellite communications.

Appendix E: Related Efforts

In addition to the commercial vendor and research communities, several organizations are tackling the wider issues of IDS interoperability, evaluation, and user education. We describe some of these efforts below.

Lincoln Laboratory (LL)

The following is quoted directly from a project-specific Lincoln Laboratory's Web site.

> The Information Systems Technology Group of MIT Lincoln Laboratory, under Defense Advanced Research Projects Agency (DARPA ITO) and Air Force Research Laboratory (AFRL/SNHS) sponsorship, is collecting and distributing the first standard corpora for evaluation of computer network intrusion detection systems. LL is also coordinating, with AFRL, the first formal, repeatable, and statistically-significant evaluations of intrusion detection systems. These evaluations measure probability of detection and probability of false-alarm for each system under test.
>
> These evaluations are contributing to the intrusion detection research field by providing direction for research efforts and an objective calibration of the current technical state-of-the-art. They are of interest to all researchers working on the general problem of workstation and network intrusion detection. The evaluation is designed to be simple, to focus on core technology issues, and to encourage the widest possible participation by eliminating security and privacy concerns, and by providing data types that are used commonly by the majority of intrusion detection systems.

This work represents the most comprehensive evaluation of research ID systems that has been performed to date. While the work is flawed in many respects, it does provide a basis for making a rough comparison of existing systems under a common set of circumstances. Only limited descriptions of the experiment have appeared in print and it may be that a more detailed exposition of the work will alleviate some of the criticisms.

The most detailed description available at the time of this report is Kristopher Kendall's BS/MS Thesis [B119]. The Lincoln Labs team has made presentations on the experiment at various meetings attended by one of the authors of this report.

These include the August, 1999 DARPA PI meeting in Phoenix, AZ and the Recent Advances in Intrusion Detection Workshop (RAID 99) at Purdue University in September, 1999.

Presentations [B120, B121] similar to ones given at those meetings appear at the Lincoln Labs experiment site, http://ideval.ll.mit.edu[1]. (Please note that this site is password-protected). A paper describing some aspects of the 1998 experiment [B144] was scheduled to be published in January, 2000.

For a variety of political, legal, and technical reasons, it was determined that actual network data could not be used for the evaluation; synthetic data was generated from which data was derived for use in the evaluation. A network was set up that generated traffic said to be similar to that seen at the network boundary of a typical Air Force base. This traffic consisted of synthetic background traffic to which attacks on a few machines inside the base were added. The data consisted of tcpdump data sniffed just outside the base, Sun Solaris BSM audit data from an attack victim host inside the base, and file system data from the host (for integrity checking). The data was made available to those who were evaluated in 1998. In 1999, additional tcpdump data from inside the base and audit data from a victim running Windows NT were provided.

There are a number of unanswered questions concerning the background data. Although it is claimed to be similar to actual data in terms of a variety of statistical characteristics, there is no evidence to demonstrate that the false alarm behavior that it elicits from the systems being evaluated is similar to the behavior they would demonstrate using real data. Since false alarms are a key measure of the evaluation, this is a critical failing. The traffic load or data rate used for the evaluation are not specified.[2] Although this is not a real time evaluation, data is tagged with times and it is possible that simulated data rate may affect the performance of some systems for both attack detection and false alarm behavior. The risks associated with the use of synthetic data in experimental settings are well known. The customary risk mitigation methods, such as the performance of pilot studies and targeted comparisons between the responses to synthetic and natural data, appear not to have been applied in this case.

A common problem in the conduct of any experimental evaluation is the choice of an appropriate unit of analysis that can be used to express the results. This needs to be chosen with care to avoid experimental bias. In 1998, the unit of analysis was chosen to be a "session" or a run[3] of one of the protocols represented in the tcpdump data.

---

1. This site is password protected. For information concerning access, contact intrusion@sst.ll.mit.edu.

2. We were given raw file sizes for 2 days of inside and 1 day of outside data for 1999. These correspond to average data rates of 34 to 48 killobits per second or less depending on the tcpdump header overhead. We do not know how this compares to average rates for the typical AFB.

3. In the tcpdump data, a session is represented by a start time, duration, protocol, source (host and port) and destination (host and port).

The evaluatees were provided with some of weeks of "training" data in which all sessions were identified and sessions containing an attack were labeled with the attack they contained.

Evaluation data was similarly divided into sessions and evaluatees were asked to label each session with a measure that indicated the degree of confidence that they had in the detection of an attack in that session. Whether a measure other than 0 (NO) or 1 (YES) is appropriate depends to a large extent on the detection and evaluation approach used in the system under test.

The use of "session" as the unit of analysis may be questionable. For the attacks that are inserted into the test data, the attack is likely to be embedded in a session, although complex, multistage attacks can be spread over multiple sessions involving multiple sources and protocols. It is not the case that every message or packet that is part of an attack-containing session is part of the attack. Similarly, it is possible for a system to raise a false alarm based on examination of messages from several distinct sessions. This is a hard problem and it is not clear what unit of analysis is appropriate, but whatever choice is made should be justified as a part of the experiment design.

The results of the evaluation are presented as Receiver Operating Curves (ROC) which may or may not be appropriate for all systems. The ROC plots percentage of attacks detected against percentage of false alarms. Under the proper circumstances, it is a powerful method for presenting system performance in a way that separates system behavior from environmental factors. If a binary measure of detection confidence is used, the ROC will consist of a single operating point. Lines are usually drawn from the (0,0) point to the operating point and from the operating point to the (1,1) point on the graph. This is based on the assumption that the decision criteria that produced the point could be changed to move the decision towards the point of universal rejection (no sessions are intrusions) with 0% detection and 0% false alarms or towards the point of universal acceptance (all sessions are intrusions) with 100% detection and 100% false alarm rate. For example, intuitively, this does not seem applicable to rule-based systems that have no memory. If the measure varies, then a curve can be obtained by varying a threshold and counting as detections only those whose measures exceed the threshold. These curves are meaningful if the measure is based on some knowledge of the distribution of attacks in the background. It was the intent of the "training data" to provide this knowledge, but this approach raises additional questions about the applicability of training based on artificial data to the real world[1] and on the feasibility of performing such training in the real world.

1. Both the frequency and distribution of attack types contained in the evaluation data seems to be substantially different from what we have observed in the wild. We know of no published attack data for the sites on which the evaluation data was modeled, but suspect that the evaluation data is not typical in this respect.

For 1999, no unit of analysis is specified and ROC-like curves with an X axis giving false alarms per day [1](rather than percent false alarms) is used.

Detections are still reported as percent detections based on the number of attacks injected. Curves of this form appear in the keyword recognition area of speech perception, but we have been unable to discover a rationale for their use. Unless there is a relatively constant relationship between errors per unit time and percent errors, this presentation introduces substantial environmental bias into the results.[2]

In summary, the Lincoln Labs evaluation represents a monumental but incomplete effort. Many questions about the details of the evaluation design and its implementation are not answered in the published literature. The way in which the results are presented appears to be questionable, and it is not clear that the training methodology used with systems that require training can be replicated in actual deployment.

International Computer Security Association (ICSA)

The ICSA Intrusion Detection Systems Consortium (IDSC) was established in 1998 to provide an open forum in which ID product developers could work toward common goals such as educating end users, creating industry standards, achieving product interoperability, and maintaining product and marketing integrity. Any commercial vendor of intrusion detection or vulnerability assessment products and services are welcome to join this association.

The mission of the IDSC is to facilitate the adoption of intrusion detection products by defining common terminology, increasing market awareness, maintaining product integrity, and influencing industry standards. [B23, http://www.icsa.net/services.consortia/intrusion]

System Administrator and Network Security (SANS) Organization

SANS has established ID'Net, a test environment within which IDS product developers can demonstrate their products. The goal of ID'Net is to showcase all of the available IDS systems and their abilities in a real-time, controlled environment to a target audience, collect valuable data, and be an effective learning and teaching tool for all who participate. ID'Net consists of a

1. False alarms per day appear overlaid on the percent false alarm scales for some of the ROCs presented for 1999. Even here, this usage should be accompanied by caveats indicating that these figures are a function of the data rate and its composition as well as the system under evaluation.

2. Simply varying the rate at which the data is presented to the system by a factor of 10 in either direction would change the false alarm rate by a factor of 100 in the absence of rate dependent sensitivities in the evaluated system. Under the same circumstances, the true detection percentage would remain constant.

DNS server and a SMTP server running Linux and a Windows NT-based web server. Other boxes on the net include vendor and SANS ID systems.

Tests performed within ID'Net include scanning for vulnerabilities and inviting other systems to launch attacks against the environment in which the products are installed. At the May, 1999 SANS Conference, seven vendors participated by installing their ID products and then monitoring how successfully those products scanned for eleven types of vulnerabilities and detected nineteen types of attacks. SANS plans to enhance the test environment's capabilities and invite other vendors to participate at future SANS conferences.

For more description on the next three efforts, refer to *Network Intrusion Detection* by Steven Northcutt [B76].

Open Platform for Secure Enterprise Connectivity (OPSEC Checkpoint)

OPSEC has been around for over a year and is stable and widely used as an application programming interface (API). The API is published and available as a software development kit. Further information is available at http://www.checkpoint.com/opsec/index.html.

Common Content Inspection (CCI — Checkpoint)

CCI is also an API. Once the firewall or IDS sensor has grabbed the packet, file, or communication stream and realized that it needs additional inspection, it redirects it to an inspection engine within CCI. The types of inspections that are performed include intrusion signatures, prohibited URLs, viruses, hostile applets, and scanning for content that is company proprietary. Further information is available at http://www.stardust.com/cciapi/docs/010799/CCIAPIScopeDraft3011.doc.

Adaptive Network Security Alliance (ANSA — ISS)

ANSA allows vendors to create systems that work with and enhance the ISS family of products. This interoperability specification will support four functional areas: automated response, lockdown, decision support, and security management. Further information is available at http://ansa.iss.net.

Emerging Standards

Within the past few years, the ID community has taken action to create standards for the communication of intrusion data between different ID components.

Having standards to communicate intrusion data would enable ID systems from multiple vendors to combine and inter-operate, thus forming a more complete and comprehensive IDS. There are two significant efforts underway to define standards for the communication and exchange of intrusion data.

Common Intrusion Detection Framework (CIDF) (Research Community Pursuit)

The CIDF project, which is sponsored by DARPA, is developing protocols and APIs to enable ID research projects to share data and resources. CIDF is not intended as a standard that will influence the commercial marketplace; it is a research project. In addition to defining protocols for communication between ID systems, this project has also created a Common Intrusion Specification Language (CISL) that defines a standard way to represent intrusion data. Further information about CIDF is available at http://gost.isi.edu/cidf.

Intrusion Detection Working Group (IDWG) (Primarily Vendor Community Pursuit)

The IDWG is an Internet Engineering Task Force working group which was formed by vendors in the ID community who did not like some of the work done by the CIDF. As stated in its charter, this working groups intends to "define data formats and exchange procedures for sharing information of interest to ID and response systems, and to management systems that may need to interact with them." The charter and mailing list archive for the IDWG are available at http://www.ietf.org/html.charters/idwg-charter.html and at http://www.semper.org/idwg-public>, respectively.

Appendix F: Candidate IDS Selection Criteria

The information in this appendix is taken from a paper by Edward Amoroso and Richard Kwapniewski titled "A Selection Criteria for Intrusion Detection Systems" [B57]. This paper includes a questionnaire which can be used independently or sent to vendors to determine the detection and operational capabilities of an IDS. Using this questionnaire, each area of the IDS is rated using three categories, listed below from least to most capable.

Detection Capabilities:

- Time to detected intrusions — elapsed time between an intrusive event and its automated detection in three categories from least capable to most capable:
  a. eventually detect intrusions, but perhaps well after the attack has completed
  b. more timely on-the-fly network information capture for prompt reporting of intrusions (minutes to hours)
  c. specially designed functionality to minimize time of detections using, e.g., hardware-based packet capture and processing solutions
- Stored knowledge about intrusions — the richness and extensibility of the intrusion knowledge base used in the product; includes the amount of work required to encode a new detection approach, implement it in a product, and integrate the new product feature into an embedded base. The three categories are
  a. some set of intrusion profiles and content patterns is provided by the vendor and updated periodically
  b. content patterns can be customized by the administrator
  c. provide an API or language for users to specify customized intrusion profiles for non-trivial attack patterns
- User and system activity profiling — the ability to develop customer profiles of users and systems as the basis for detecting anomalous behavior. The three categories are
  a. no ability to develop profiles
  b. the existence of a profiling engine for specifying and capturing broad statistical information about system operation

c. profiling granularity increased considerably to that individual user profiles can be used with administrator-tunable thresholds

- Multi-source information correlation — the ability of an IDS to relate monitored traffic or alarms from different sources and at different times. The three categories are

  a. functionality must exist for multiple feeds of information to be collected in a common processing location for display and minimal response

  b. include tools to define and search for patterns of activity in the monitored traffic feeds

  c. include an automated mechanism for incorporating out-of-band sources such as detected events from other networks, information pulled from news feeds, or real time operator generated input into the — correlation processing; also provides some sort of API for accepting alarm message from out-of-band sources

Operational Capabilities:

- Test and Verification The three categories from least capable to most capable are

  a. no test and verification

  b. vendor supplies evidence that the system has been appropriately tested and verified using a well-defined test suite

  c. vendor provides evidence of third-party testing by an independent organization that will supply documentation of test plans and results

- System Security — the degree to which the IDS protects its resources from malicious security attack. The three categories are

  a. provide basic system security for its platforms including access control, some level of audit, and user authentication

  b. the presence of an encryption-based VPN capability for protecting communications

  c. the presence of a security hardened base for servers and special functionality and network configurations that are designed to enhance the stealth nature of the system

- Supported Network Media — indicates the types of network media on which the system can be connected at the required bandwidth without dropping any packets. The three categories are

  a. supports low-speed LAN and WAN interfaces (up to 16 Mb/s) such as Ethernet, token Ring, NxDS0 and DS1.

  b. supports high-speed LAN and WAN interfaces (up to 100 Mb/s) such as fast Ethernet, FDDI, and DS3.

c. supports very high-speed media (over 100 Mb/s) such as Gigabit Ethernet, ATM OC-3, and ATM OC-12.

- User Interface — the type of interface provided to the IDS administrator for configuration and monitoring components. The three categories are

  a. no formal interface; relies on resources available from the native operating system on which it resides

  b. provide a dedicated command-line or GUI application which can be used to perform all necessary operations

  c. provide a centralized GUI application from which an administrator can remotely control and operate any component of the IDS

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 2000	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
State of the Practice of Intrusion Detection Technologies	C — F19628-95-C-0003

6. AUTHOR(S)
Julia Allen, Alan Christie, William Fithen, John McHugh, Jed Pickel, Ed Stoner

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU/SEI-99-TR-028

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ESC-99-028

11. SUPPLEMENTARY NOTES

12.a DISTRIBUTION/AVAILABILITY STATEMENT	12.b DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (maximum 200 words)

Attacks on the nation's computer infrastructures are a serious problem. Over the past 12 years, the growing number of computer security incidents on the Internet has reflected the growth of the Internet itself. Because most deployed computer systems are vulnerable to attack, intrusion detection (ID) is a rapidly developing field. Intrusion detection is an important technology business sector as well as an active area of research. Vendors make many claims for their products in the commercial marketplace so separating hype from reality can be a major challenge. A goal of this report is to provide an unbiased assessment of publicly available ID technology. We hope this will help those who purchase and use ID technology to gain a realistic understanding of its capabilities and limitations. The report raises issues that we believe are important for ID system (IDS) developers to address as they formulate product strategies. The report also points out relevant issues for the research community as they formulate research directions and allocate funds.

14. SUBJECT TERMS	15. NUMBER OF PAGES
intrusion detection, intrusion detection systems, intrusion detection technologies, IDS, computer security, information security, network security	220
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL